



TUGAS AKHIR - KI141502

**RANCANG BANGUN APLIKASI *VIRTUAL MUSIC COMPOSER*, SEBAGAI MEDIA
MENCIPTAKAN SEBUAH LAGU
MENGUNAKAN LEAP MOTION CONTROLLER**

Angga Saputra Dwi Wardana
NRP 5112100085

Dosen Pembimbing
Dr. Darlis Herumurti, S.Kom., M.Kom.
Ridho Rahman Hariadi, S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016



FINAL PROJECT - KI141502

DESIGN AND IMPLEMENTATION OF VIRTUAL MUSIC COMPOSER APPLICATION, AS A MEDIA FOR PRODUCE A SONG USING LEAP MOTION CONTROLLER

Angga Saputra Dwi Wardana
NRP 5112100085

Advisor

Dr. Darlis Herumurti, S.Kom., M.Kom.
Ridho Rahman Hariadi, S.Kom., M.Sc.

DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016

LEMBAR PENGESAHAN

**Rancang Bangun Aplikasi *Virtual Music Composer*,
Sebagai Media Menciptakan Sebuah Lagu Menggunakan
Leap Motion Controller**

TUGAS AKHIR

**Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Interaksi Grafis dan Seni
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember**

Oleh :

**Angga Saputra Dwi Wardana
NRP : 5112100085**

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Dr. DARLIS HERUMURTI, S.Kom., M.Kom.
NIP: 197712172003121001

(pembimbing 1)

RIDHO RAHMAN HARIADI, S.Kom., M.Sc.
NIP: 198603122012122004

(pembimbing 2)

**SURABAYA
JULI 2016**

RANCANG BANGUN APLIKASI *VIRTUAL MUSIC COMPOSER*, SEBAGAI MEDIA MENCIPTAKAN SEBUAH LAGU MENGGUNAKAN LEAP MOTION CONTROLLER

Nama Mahasiswa : Angga Saputra Dwi Wardana
NRP : 5112100085
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing 1 : Dr. Darlis Herumurti, S.Kom., M.Kom.
Dosen Pembimbing 2 : Ridho Rahman Hariadi, S.Kom., M.Sc.

ABSTRAK

Teknologi pendeteksi gesture tangan dan alat saat ini berhasil dikembangkan dengan baik pada perangkat lunak bernama Leap Motion Controller. Alat ini sendiri berfungsi sebagai alternatif dalam input control dalam interaksi antara manusia dengan komputer tanpa menggunakan sentuhan.

Ide yang digunakan dalam tugas akhir ini akan dibangun sebuah aplikasi simulasi memainkan alat musik drum dan terompet dan dipadukan dengan bernyanyi, hingga akhirnya ketiga suara tersebut digabungkan menjadi sebuah lagu. Aplikasi ini dibangun menggunakan Unity dengan Leap Motion SDK. Tujuannya diharapkan pengguna dapat merasakan sensasi bermain drum dan terompet menyerupai aslinya.

Aplikasi telah diuji oleh 8 orang. Aplikasi memiliki desain antarmuka yang baik dengan penilaian 86% dan tingkat kenyamanan dengan penilaian 73,4%. Namun, tingkat immersive kurang baik dengan penilaian 68%. Hal ini dikarenakan bagian-bagian pada drum memiliki ketinggian yang berbeda, sehingga pengguna harus menebak ketinggian tersebut sebelum memukul dan deteksi pergerakan jari kurang sensitif untuk memainkan terompet.

Kata kunci: Leap Motion, Unity, drum virtual, terompet virtual

Design And Implementation Of Virtual Music Composer Application, As A Media For Produce A Song Using Leap Motion Controller

Student Name : Angga Saputra Dwi Wardana
Student ID : 5112100085
Major : Department Of Informatics FTIf-ITS
Advisor 1 : Dr. Darlis Herumurti, S.Kom., M.Kom.
Advisor 2 : Ridho Rahman Hariadi, S.Kom, M.Sc.

ABSTRACT

Hand gesture and tools reader technology is now successfully well developed on a device called Leap Motion which serves as an alternative input control in the interaction between humans and computers without a touch.

The idea used in this final project will be built a simulation application plays drums and trumpets and combined with singing, until finally the third sound is produced into a song. This application is built using Unity with Leap Motion SDK. The goal is expected the user can feel the sensation of playing drums and trumpet like the original.

Application has been tested by 8 people. The application has a good interface design with 86% votes and a comfort level with 73.4% votes. However, the level of immersive less well with 68% votes. This is because the parts on the drums have different heights, so users have to guess at these heights before hitting and finger movement detection is less sensitive to play the trumpet.

Keywords: Leap Motion, Unity, virtual drum, virtual trumpet

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxiii
DAFTAR KODE SUMBER	xxv
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Permasalahan.....	2
1.3. Batasan Permasalahan	2
1.4. Tujuan.....	3
1.5. Manfaat.....	3
1.6. Metodologi	3
1.6.1. Penyusunan proposal Tugas Akhir	4
1.6.2. Studi literatur	4
1.6.3. Analisis dan Perancangan Sistem	5
1.6.4. Implementasi	5
1.6.5. Pengujian dan Evaluasi	5
1.6.6. Penyusunan Buku Tugas Akhir	5
1.7. Sistematika Penulisan.....	6
BAB II DASAR TEORI.....	9
2.1. Rancang Bangun Perangkat Lunak	9
2.2. Virtual Music Composer	9
2.3. Permodelan Tiga Dimensi (3D)	10
2.4. Leap Motion Controller	10
2.5. SDK Leap Motion Developer.....	13
2.5.1. Pointables	13
2.5.2. <i>InteractionBox</i>	14
2.6. Unity 3D	15
2.7. Visual Studio	16

2.8.	Blender	17
2.9.	Wavepad Sound Editor	18
2.10.	Bandicams	18
2.11.	Tools Detection pada Aplikasi Virtual Indonesian Music Instrument (VIMi)	19
2.12.	Finger Motion pada Aplikasi Realitas Virtual Pertunjukan Wayang Golek.....	20
BAB III ANALISIS DAN PERANCANGAN SISTEM.....		21
3.1.	Analisis Perangkat Lunak	21
3.1.1.	Deskripsi Umum Perangkat Lunak.....	21
3.1.2.	Spesifikasi Kebutuhan Fungsional Perangkat Lunak	22
3.1.3.	Identifikasi Pengguna	23
3.2.	Perancangan Perangkat Lunak.....	23
3.2.1.	Model Kasus Penggunaan	23
3.2.2.	Definisi Kasus Penggunaan	24
3.2.3.	Definisi Aktor	29
3.2.4.	Perancangan Model 3D	30
3.2.5.	Arsitektur Umum Sistem	32
3.2.6.	Rancangan Antarmuka Aplikasi.....	33
3.2.7.	Perancangan Proses Aplikasi	39
BAB IV IMPLEMENTASI		45
4.1.	Lingkungan Implementasi	45
4.2.	Implementasi Antar muka	46
4.2.1.	Impelementasi Antarmuka Menu Utama.....	46
4.2.2.	Impelementasi Antarmuka Drum	46
4.2.3.	Impelementasi Antarmuka Terompet	47
4.2.4.	Implementasi Antarmuka Mikrofon	48
4.2.5.	Impelementasi Antarmuka Melihat Bantuan	48
4.3.	Implementasi Proses	52
4.3.1.	Implementasi Melihat Bantuan.....	53
4.3.2.	Implementasi Memainkan Drum	57
4.3.3.	Implementasi Memainkan Suara Terompet.....	60
4.3.4.	Implementasi Menyanyikan Sebuah Lagu.....	70
4.3.5.	Implementasi Memainkan Hasil Rekaman	71

4.4.	Implementasi Integrasi Leap Motion dengan Unity	72
BAB V PENGUJIAN DAN EVALUASI		73
5.1.	Lingkungan Pengujian.....	73
5.2.	Pengujian Aplikasi	74
5.2.1.	Skenario Pengujian Fungsionalitas.....	74
5.2.2.	Hasil Uji Coba Fungsionalitas.....	74
5.3.	Pengujian Pengguna	85
5.3.1.	Skenario Pengujian Pengguna	85
5.3.2.	Daftar Penguji Perangkat Lunak	87
5.3.3.	Hasil Pengujian Pengguna	87
5.3.4.	Kritik dan Saran Pengguna	91
5.4.	Evaluasi Pengujian	92
BAB VI KESIMPULAN DAN SARAN.....		95
6.1.	Kesimpulan.....	95
6.2.	Saran.....	96
DAFTAR PUSTAKA.....		97
LAMPIRAN		99
BIODATA PENULIS.....		107

DAFTAR GAMBAR

Gambar 2.1 Perkembangan Leap Motion Cntroller	11
Gambar 2.2 Leap Motion Controller.....	11
Gambar 2.3 Area Jangkauan Inframerah dalam Dua Dimensi	12
Gambar 2.4 Area Jangkauan Inframerah dalam Tiga Dimensi ...	12
Gambar 2.5 Representasi Finger	14
Gambar 2.6 Representasi Tool	14
Gambar 2.7 Representasi Wilayah dari InteractionBox	15
Gambar 2.8 Tampilan Antarmuka Unity 3D	16
Gambar 2.9 Tampilan Antarmuka Visual Studio	17
Gambar 2.10 Tampilan Antarmuka Blender	17
Gambar 2.11 Tampilan Wavepad Sound Editor.....	18
Gambar 2.12 Tampilan Antarmuka Bandicams	19
Gambar 3.1 Diagram Kasus Penggunaan	24
Gambar 3.2 Perancangan Model 3D Drum Tampak Atas.....	31
Gambar 3.3 Perancangan Model 3D Terompet Tampak Samping	31
Gambar 3.4 Perancangan Model 3D Mikروفon	32
Gambar 3.5 Rancangan Sederhana Arsitektur Aplikasi	32
Gambar 3.6 Rancangan Antarmuka Menu Utama	33
Gambar 3.7 Rancangan Antarmuka Drum	Error! Bookmark not defined.
Gambar 3.8 Rancangan Antarmuka Terompet.....	36
Gambar 3.9 Rancangan Antarmuka Mikروفon	37
Gambar 3.10 Rancangan Antarmuka Melihat Bantuan.....	39
Gambar 3.11 Flowchart Rancangan Proses Memainkan Drum ..	40
Gambar 3.12 Flowchart Rancangan Proses Memainkan Terompet	41
Gambar 3.13 Flowchart Rancangan Proses Menyanyikan Sebuah lagu	42
Gambar 3.14 Flowchart Rancangan Proses Memainkan Hasil Rekaman.....	43
Gambar 4.1 Implementasi Antarmuka Menu Utama	46
Gambar 4.2 Implementasi Antarmuka Drum	47

Gambar 4.3 Implementasi Antarmuka Terompet.....	47
Gambar 4.4 Implementasi Antarmuka Mikrofon	48
Gambar 4.5 Implementasi Antarmuka Melihat Bantuan Tahap Pertama.....	49
Gambar 4.6 Implementasi Antarmuka Melihat Bantuan Tahap Kedua.....	50
Gambar 4.7 Implementasi Antarmuka Melihat Bantuan Tahap Ketiga	51
Gambar 4.8 Implementasi Antarmuka Melihat Bantuan Tahap Keempat.....	52
Gambar 4.9 Implementasi Antarmuka Melihat Bantuan Tahap Kelima	52
Gambar 4.10 Grafik Pitch Value Dari Tiupan Pengguna	62
Gambar 4.11 Grafik Pitch Value Dari Noise Lingkungan	62
Gambar 4.12 Gambar Tulang dan Sendi Pada Jari.....	67
Gambar 4.13 Penentuan Selisih Nilai Y	67
Gambar 4.14 Hasil Proses Impor <i>Package Leap Motion</i>	72
Gambar 4.15 Implementasi Input Area Leap Motion.....	72
Gambar 5.1 Stik Biasa Dalam Keadaan Tidak Menyilang.....	75
Gambar 5.2 Stik Biasa Dalam Keadaan Menyilang	76
Gambar 5.3 Stik Diberi Isolasi Dalam Posisi Tidak Menyilang ..	76
Gambar 5.4 Stik Diberi Isolasi Dalam Posisi Menyilang.....	77
Gambar 5.5 Uji Coba Saat Memainkan Drum	78
Gambar 5.6 Uji Coba Saat Merekam Drum	79
Gambar 5.7 Uji Coba Saat Memainkan Terompet	80
Gambar 5.8 Uji Coba Saat Merekam Terompet	81
Gambar 5.9 Uji Coba Saat Menyanyikan Sebuah Lagu	82
Gambar 5.10 Uji Coba Saat Merekam Mikrofon	83
Gambar 5.11 Grafik Penilaian Antarmuka	90
Gambar 5.12 Grafik Penilaian Tingkat <i>Immersive</i>	90
Gambar 5.13 Grafik Penilaian Tingkat Kenyamanan.....	91
Gambar A.0.1 Kuesioner Responden Pertama	99
Gambar A.0.2 Kuesioner Responden Kedua.....	100
Gambar A.0.3 Kuesioner Responden Ketiga.....	101
Gambar A.0.4 Kuesioner Responden Keempat.....	102

Gambar A.0.5 Kuesioner Responden Kelima	103
Gambar A.0.6 Kuesioner Responden Keenam	104
Gambar A.0.7 Kuesioner Responden Ketujuh	105
Gambar A.0.8 Kuesioner Responden Kedelapan	106

DAFTAR TABEL

Tabel 3.1 Daftar Kasus penggunaan.....	24
Tabel 3.2 Spesifikasi Kasus Merekam dan Memainkan Drum ...	25
Tabel 3.3 Spesifikasi Kasus Merekam dan Memainkan Terompet	26
Tabel 3.4 Spesifikasi Kasus Penggunaan Merekam dan Menyanyikan Sebuah Lagu	27
Tabel 3.5 Spesifikasi Kasus Penggunaan Memainkan Hasil Rekaman	28
Tabel 3.6 Spesifikasi Kasus Penggunaan Melihat Bantuan	29
Tabel 3.7 Deskripsi Pengguna.....	30
Tabel 3.8 Spesifikasi Atribut Antarmuka Menu Utama	34
Tabel 3.9 Spesifikasi Atribut Antarmuka Drum	Error! Bookmark not defined.
Tabel 3.10 Spesifikasi Atribut Antarmuka Terompet	36
Tabel 3.11 Spesifikasi Atribut Antarmuka Mikrofon.....	38
Tabel 3.12 Spesifikasi Atribut Antarmuka Melihat Bantuan	39
Tabel 4.1 Spesifikasi Lingkungan Implementasi	45
Tabel 5.1 Lingkungan Pengujian Sistem.....	73
Tabel 5.2. Hasil Uji Coba Merekam dan Memainkan Drum.....	77
Tabel 5.3 Hasil Uji Coba Merekam dan Memainkan Terompet ..	79
Tabel 5.4 Uji Coba Menyanyikan Sebuah Lagu	81
Tabel 5.5 Uji Coba Merekam Setiap Alat	83
Tabel 5.6 Uji Coba Melihat Bantuan.....	84
Tabel 5.7 Rentang Nilai.....	86
Tabel 5.8 Kuesioner Pengguna.....	86
Tabel 5.9 Daftar Penguji Perangkat Lunak	87
Tabel 5.10 Hasil Pengujian Pengguna.....	88
Tabel 5.11 Hasil Akhir Pengujian Pengguna	89
Tabel 5.12 Kritik dan Saran Pengguna.....	91

DAFTAR KODE SUMBER

Kode Sumber 4.1 Melihat Bantuan	56
Kode Sumber 4.2 Mendeteksi Stik	58
Kode Sumber 4.3 Memainkan Drum	59
Kode Sumber 4.4 Merekam Suara Drum	60
Kode Sumber 4.5 Mendeteksi Tiupan	64
Kode Sumber 4.6 Mendeteksi Telapak Tangan	66
Kode Sumber 4.7 Mendeteksi Jari yang Sedang Menekan	69
Kode Sumber 4.8 Merekam Suara Terompet	69
Kode Sumber 4.9 Menyanyikan Sebuah Lagu	70
Kode Sumber 4.10 Memainkan Hasil Rekaman	71

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

1.1. Latar Belakang

Interaksi Manusia dan Komputer (IMK) adalah bidang penting dalam perkembangan teknologi saat ini yang menjadi pusat perhatian sejak munculnya perangkat baru yang memanfaatkan *motion capture* untuk menunjang interaksi dengan komputer (seperti Nitendo Wii Remote, Microsoft Kinect dan Leap Motion) [1]. Penggunaan antarmuka gestural dapat meningkatkan keuntungan dan kemudahan terlebih interaksi menjadi bagian penting dalam aplikasi seperti seni, bantuan medis, simulasi, dan *music production*. Leap Motion yang nantinya akan digunakan dalam tugas akhir ini merupakan alat penerima sensor gerakan jarak dekat. Alat ini digunakan untuk menangkap *input* dan kontrol terhadap komputer tanpa menyentuh sama sekali.

Musik komputer telah berkembang dan menghasilkan berbagai macam Digital Music Instruments (DMIs). DMIs merupakan instrumen musik yang dihasilkan dari alat elektronik atau aplikasi komputer yang dapat mengeluarkan suara melalui *speaker* layaknya suara asli dari alat musik tersebut. DMIs saat ini sudah banyak digunakan di sebagian besar gaya musik internasional. Penelitian terhadap perkembangan DMIs menjadi sangat aktif terlebih banyak seniman internasional yang memanfaatkan DMIs untuk menghasilkan sebuah karya [1]. Dengan adanya DMIs dapat memudahkan seniman dalam menciptakan karya musik karena hanya dengan aplikasi komputer seorang seniman dapat memainkan berbagai macam alat musik tanpa harus memiliki alat tersebut. Namun sedikit dari DMIs yang memanfaatkan teknologi *hand gesture* sebagai *input*.

Dalam aplikasi yang akan dibuat, pengguna dapat menghasilkan DMIs dengan memainkan drum dan terompet secara virtual. Dengan menggunakan Leap Motion pengguna diharapkan dapat merasakan sensasi nyata bermain alat musik tersebut. Leap motion akan mendeteksi jari pengguna atau stik kayu sebagai alat tambahan, lalu posisi jari atau stik akan ditampilkan dalam bentuk virtual dalam aplikasi dengan pergerakan mengikuti gerak jari atau stik pengguna. Ketika jari atau stik pengguna melakukan gerakan atau pukulan, lalu gambaran virtual dalam aplikasi juga akan melakukan gerakan atau pukulan. Nantinya suara akan dihasilkan dari gerakan atau pukulan yang dilakukan oleh pengguna. Suara yang dihasilkan dari setiap alat musik dapat direkam dan disatukan hingga menjadi suatu instrumen yang harmoni. Pengguna dapat menambahkan suaranya dengan menggunakan mikrofon sebagai pelengkap instrumen yang telah dihasilkan sehingga sebuah lagu dapat tercipta melalui aplikasi ini.

1.2. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana cara membuat aplikasi *music composer* untuk menciptakan sebuah lagu dengan menggunakan virtual drum dan virtual terompet sebagai alat musiknya?
2. Bagaimana cara merancang virtual drum pada aplikasi ini sehingga dapat dimainkan menyerupai aslinya?
3. Bagaimana cara merancang virtual terompet pada aplikasi ini sehingga dapat dimainkan menyerupai aslinya?

1.3. Batasan Permasalahan

Beberapa batasan masalah yang terdapat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Aplikasi yang dibuat merupakan aplikasi berbasis desktop.
2. Lingkungan pengembangan yang digunakan menggunakan aplikasi Unity 3D dengan bahasa pemrograman C#.

3. Aplikasi dibuat menggunakan Leap Motion SDK.
4. Alat musik yang ditampilkan merupakan representasi 3D dari alat musik tersebut.
5. Alat tambahan yang digunakan dalam menggunakan aplikasi ini adalah stik drum dan pedal yang disediakan untuk memainkan drum serta mikrofon untuk bernyanyi dan menerima tiupan dari pengguna untuk membunyikan terompet.

1.4. Tujuan

Tujuan dari pembuatan tugas akhir ini antara lain :

1. Pengembangan Leap Motion sebagai alat bantu interaksi dalam industri musik.
2. Membangun perangkat lunak yang dapat mensimulasikan cara bermain alat musik drum dan terompet.
3. Membangun perangkat lunak yang dapat menghasilkan sebuah lagu.

1.5. Manfaat

Manfaat yang diharapkan dari pengembangan aplikasi ini diantaranya:

1. Terciptanya sebuah aplikasi yang dapat mensimulasikan cara bermain alat musik drum dan terompet dengan memanfaatkan Leap Motion.
2. Dapat menjadi referensi untuk penelitian lebih lanjut mengenai pengembangan leap motion dalam bidang musik.

1.6. Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

1.6.1. Penyusunan proposal Tugas Akhir

Proposal tugas akhir ini berisi mengenai rencana pengembangan aplikasi *virtual music composer* yaitu aplikasi untuk menciptakan sebuah lagu dengan memanfaatkan Leap Motion Controller. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat dari hasil pembuatan tugas akhir. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Sub bab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula subbab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

1.6.2. Studi literatur

Tahap studi literatur merupakan tahap pembelajaran dan pengumpulan informasi yang digunakan untuk mengimplementasikan Tugas Akhir. Tahap ini diawali dengan pengumpulan literatur, diskusi, eksplorasi teknologi dan pustaka, serta pemahaman dasar teori yang digunakan pada topik Tugas Akhir. Literatur-literatur yang dimaksud disebutkan sebagai berikut:

- a. Computer Music Instrumen
- b. Pemodelan Tiga Dimensi (3D)
- c. Leap Motion Controller
- d. SDK Leap Motion Developer
- e. Unity 3D
- f. Visual Studio
- g. Blender
- h. Wavepad Sound Editor
- i. Bandicams

1.6.3. Analisis dan Perancangan Sistem

Pada tahap ini dilakukan analisis dan pendefinisian kebutuhan sistem untuk masalah yang sedang dihadapi. Selanjutnya, dilakukan perancangan sistem dengan beberapa tahap sebagai berikut:

- a. Perancangan diagram kasus penggunaan.
- b. Perancangan model 3D.
- c. Perancangan antarmuka sistem.
- d. Perancangan proses aplikasi.

1.6.4. Implementasi

Pada tahap ini dilakukan implementasi pemodelan objek-objek 3D dan implementasi proses yang telah didefinisikan pada bab Analisis dan Perancangan Sistem. Kemudian dilakukan integrasi aplikasi dengan perangkat Leap Motion. Aplikasi ini dibangun dengan Unity.

1.6.5. Pengujian dan Evaluasi

Pada tahap ini dilakukan pengujian aplikasi kepada pengguna secara langsung. Pengujian aplikasi dilakukan untuk mengevaluasi hasil program. Uji fungsionalitas untuk mengetahui apakah aplikasi sudah memenuhi semua kebutuhan fungsional. Tahapan-tahapan dari pengujian adalah sebagai berikut:

- a. Pengujian kebutuhan fungsional.
- b. Kuesioner terhadap aplikasi dari pengguna.

1.6.6. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan pendokumentasian dan pelaporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan Tugas Akhir.

1.7. Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Dasar Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan antarmuka pada aplikasi.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan perangkat lunak.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat keluaran yang dihasilkan oleh aplikasi dan evaluasi untuk mengetahui kemampuan aplikasi serta mengetahui penilaian aspek kegunaan (*usability*) dari perangkat lunak.

Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

Lampiran

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

BAB II

DASAR TEORI

Bab ini menjelaskan tentang tinjauan pustaka yang menjadi dasar pembuatan Tugas Akhir. Beberapa teori, pustaka, dan teknologi yang mendasari pengerjaan Tugas Akhir ini diantaranya meliputi definisi *computer music instrument*, pemodelan tiga dimensi, Leap Motion Controller, SDK Leap Motion, Blender, Visual Studio, dan Wavepad Audio Editor. Penjelasan secara khusus masing-masing tinjauan pustaka dapat dilihat pada masing-masing subbab berikut ini.

2.1. Rancang Bangun Perangkat Lunak

Rancang bangun perangkat lunak suatu ilmu yang mempelajari proses pembuatan aplikasi yang melingkupi analisis permasalahan dan kebutuhan, perencanaan, analisis sistem, implementasi, serta aktivitas pengujian dan pemeliharaan perangkat lunak [2].

Rancang bangun perangkat lunak ini sendiri sangat penting dalam proses penentuan konsep - konsep yang akan di gunakan dalam proses pembuatan perangkat lunak, apabila diterapkan diharapkan perangkat lunak yang tercipta akan memiliki kualitas yang tinggi, manajemen waktu dan kerja yang tertata serta perangkat yang mudah dalam pemeliharaannya. Model - model yang digunakan diantaranya adalah model air terjun (Waterfall) dan model iterasi.

2.2. *Virtual Music Composer*

Komposer musik adalah seseorang yang menulis dan menciptakan musik asli yang digunakan untuk memproduksi berbagai jenis gaya musik. Kebanyakan seorang komposer terampil dalam memainkan satu atau beberapa instrumen dan memiliki telinga yang baik dalam mendengarkan setiap nada yang dimainkan [3]. Sedangkan *Virtual Music Composer* adalah

perangkat lunak yang memungkinkan anda untuk menjadi seorang composer dengan membuat jenis musik anda sendiri dengan cara memainkan alat musik virtual.

2.3. Permodelan Tiga Dimensi (3D)

Pemodelan adalah membentuk suatu benda-benda atau objek. Membuat dan mendesain objek tersebut sehingga terlihat seperti hidup. Sesuai dengan objek dan basisnya, proses ini secara keseluruhan dikerjakan di komputer. Melalui konsep dan proses desain, keseluruhan objek bisa diperlihatkan secara 3 dimensi, sehingga banyak yang menyebut hasil ini sebagai pemodelan 3 dimensi [4].

Pemodelan 3 dimensi memiliki beberapa aspek yang harus diperhatikan, yaitu pendeskripsian objek, tujuan dari model, tingkat kerumitan, kesuaian dan kenyamanan, serta kemudahan manipulasi objek. Pemodelan 3 dimensi dapat dilakukan dengan bantuan aplikasi-aplikasi, seperti Blender, K-3D, Google SketchUp, dan masih banyak lainnya.

2.4. Leap Motion Controller

Leap Motion Controller merupakan suatu perangkat yang dikembangkan oleh Leap Motion, Inc yang dikembangkan pada tahun 2008 oleh David Holz Leap Motion Controller sendiri digunakan sebagai *input* dari komputer tanpa sentuh, bisa dikatakan bahwa Leap Motion ini merupakan pengganti mouse, karena mempunyai tujuan dan fungsi yang sama. Pada tahun 2010, untuk pertama kalinya Leap Motion Controller ini diperkenalkan kepada publik [5]. Perkembangan dari Leap Motion Controller dapat dilihat pada Gambar 2.1.

Evolution of the Leap Motion Controller



Gambar 2.1 Perkembangan Leap Motion Controller

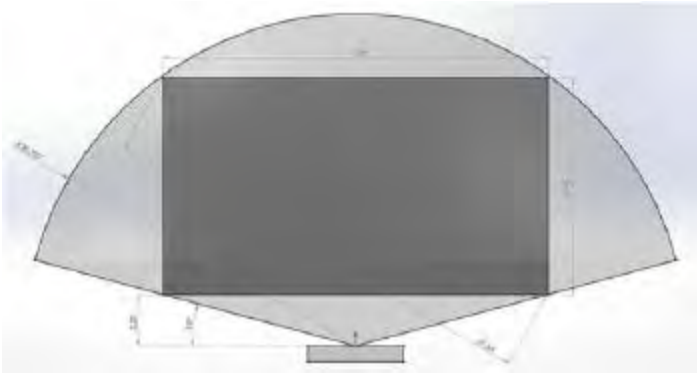
Leap Motion seperti Gambar 2.2 sendiri merupakan penemuan penting dimana alat ini adalah alat sensor perangkat keras komputer yang mendukung gerakan tangan dan jari sebagai masukan, yang dapat disamakan fungsinya seperti *mouse*, namun tidak membutuhkan kontak langsung dengan tangan atau sentuhan sehingga diharapkan adanya sebuah pengalaman baru dalam dunia *virtual*



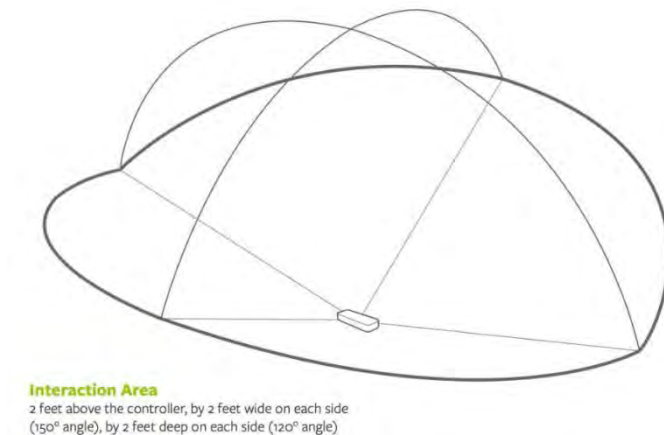
Gambar 2.2 Leap Motion Controller

Leap Motion dari sisi *hardware* sebenarnya cukup sederhana. Inti dari Leap Motion ini terletak pada pemanfaatan dua

kamera *stereo* dan terdapat tiga lampu pemancar inframerah yang menyebar secara konvergen sehingga mampu untuk menjangkau area yang lebih luas. Jadi pada tahap ini inframerah akan menyebar untuk membentuk sebuah area seperti setengah lingkaran dengan jarak jangkauan maksimal 50 cm. Area jangkauan dari Leap Motion ini dapat dilihat pada Gambar 2.3 dan Gambar 2.4 [6].



Gambar 2.3 Area Jangkauan Inframerah dalam Dua Dimensi



Gambar 2.4 Area Jangkauan Inframerah dalam Tiga Dimensi

Leap Motion Controller dapat meningkatkan interaksi manusia dengan komputer. Dalam tugas akhir ini, Leap Motion dimanfaatkan untuk melacak gerakan tangan, jari maupun suatu benda sederhana seperti stik kayu. Di saat sistem dijalankan, pengguna dapat melakukan berbagai aktivitas dengan gerakan tangan. Gerakan tangan ini diamati oleh sensor pada Leap Motion Controller sebagai input untuk melakukan berbagai hal yang dapat memicu terjadinya kejadian-kejadian dalam program. Pada simulasi alat musik drum dan terompet, gerakan stik drum yang terlacak akan menghasilkan suara seolah pengguna sedang memainkan alat musik tersebut secara langsung.

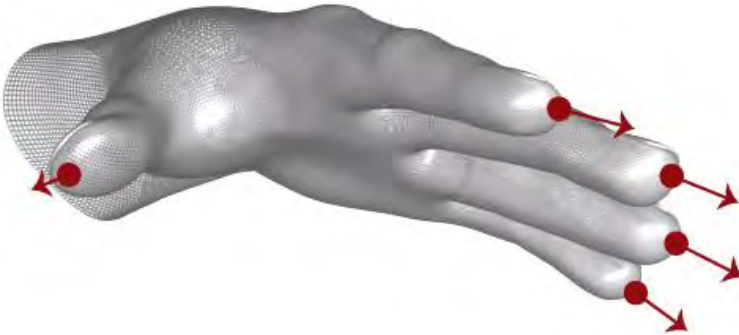
2.5. SDK Leap Motion Developer

Software Development Kit, atau yang biasa disebut SDK, adalah sebuah perangkat bantu pengembangan perangkat lunak yang digunakan untuk paket perangkat lunak tertentu, *framework* tertentu, perangkat keras tertentu, atau perangkat lain sejenisnya. Leap Motion SDK merupakan sekumpulan library yang berisikan tentang kebutuhan - kebutuhan sistem dari aplikasi yang digunakan untuk mengembangkan Leap Motion Controller ini. Biasanya terdapat beberapa fungsi yang dapat digunakan seperti pemodelan tangan dan jari - jari manusia. Leap Motion SDK ini dibuat untuk memudahkan developer dalam membangun aplikasi. Pengembang dapat mengunduhnya secara gratis dan tersedia dalam berbagai jenis bahasa pemrograman yang berbeda. Antara lain: Javascript, Unity / C#, C++, Java, Python, dan Objective-C. Juga mendukung berbagai macam sistem operasi yang berbeda yakni Windows, OSX, dan Linux.

2.5.1. Pointables

Merupakan objek yang memiliki ujung yang terdeteksi oleh Leap Motion sehingga dapat digunakan sebagai *pointing*, *screentap*, *scaling*, *rotating*, dan sebagainya. Pointables sendiri memiliki dua representasi, yakni Finger dan Tools. Adapun objek

yang terdeteksi Finger adalah objek yang tidak lurus, mempunyai persendian seperti pada Gambar 2.5. Sedangkan Tools sendiri akan terdeteksi apabila objek tersebut memiliki panjang, lurus dan berdiameter dan berwarna cerah seperti pada Gambar 2.6 [7].



Gambar 2.5 Representasi Finger

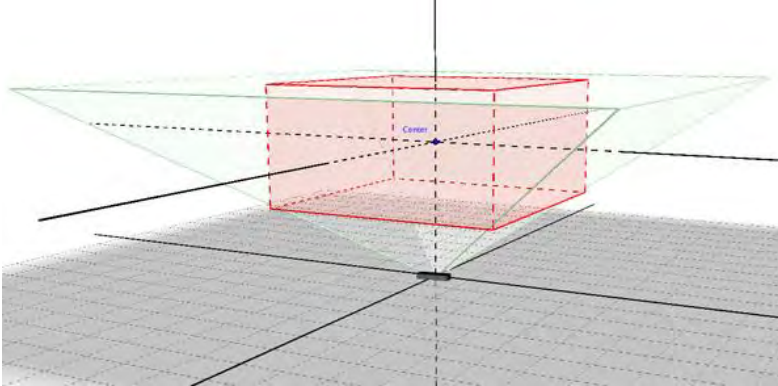


Gambar 2.6 Representasi Tool

2.5.2. *InteractionBox*

InteractionBox merupakan sebuah representasi dari sebuah bidang dalam jarak pandang Leap Motion, dimana direpresentasikan seperti sebuah kubus. Dimana hal ini digunakan untuk mempermudah penentuan koordinat dari objek baik secara dua dimensi maupun tiga dimensi yang tertangkap dalam area Leap Motion. Pada bagian ini Leap Motion merepresentasikan tiap

millimeter dengan koordinat tiga dimensi yakni x, y, dan z sesuai kebutuhan masing-masing dengan normalisasi vector [7]. *InteractionBox* dapat dilihat pada Gambar 2.7.

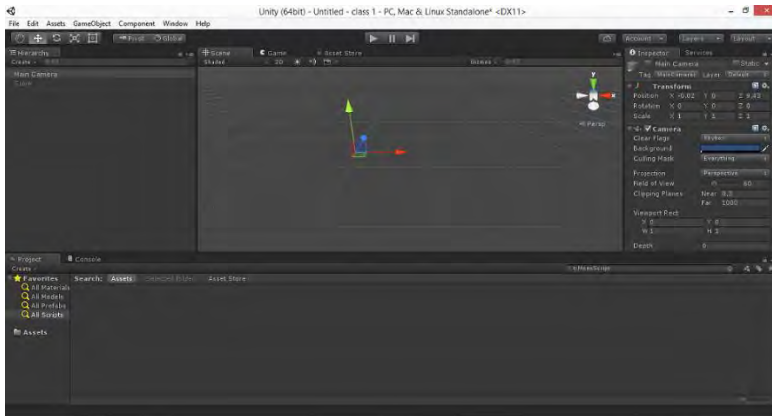


Gambar 2.7 Representasi Wilayah dari InteractionBox

2.6. Unity 3D

Unity atau Unity 3D adalah sebuah perangkat lunak yang berfungsi untuk membangun permainan atau aplikasi. Unity merupakan suatu *game development ecosystem* yang mampu digunakan untuk membuat permainan atau aplikasi dalam berbagai macam *platform* baik *console*, *desktop*, dan *mobile*. Bahasa pemrograman utama Unity adalah C# dengan IDE Mono Develop [8].

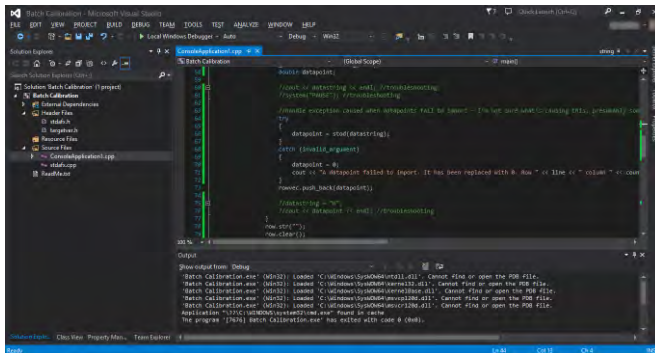
Unity 3D berbasis *cross-platform*, sehingga pengembang dapat membuat *game* yang dapat dimainkan pada perangkat komputer, ponsel pintar android, *web games* (memerlukan plugin unity *web player*), iPhone, PS3, dan bahkan X-BOX. Unity3D menyediakan *software free* dan *pro*, untuk versi gratis Unity menyediakan fitur pengembangan game berbasis windows, standalone mac dan web. Sedangkan untuk Unity *Pro* terdapat fitur yang lebih komplit dibandingkan dengan Unity Free seperti efek bayangan pada objek dan efek *water* yang lebih memukau. Tampilan antarmuka aplikasi ini dapat dilihat pada Gambar 2.8.



Gambar 2.8 Tampilan Antarmuka Unity 3D

2.7. Visual Studio

Microsoft Visual Studio merupakan sebuah aplikasi yang terdiri dari kompiler, SDK, *Integrated Development Environment (IDE)*, dan dokumentasi yang biasa digunakan untuk mengembangkan aplikasi personal, aplikasi *web*, aplikasi *mobile*, aplikasi konsole maupun komponen aplikasi pada Microsoft Visual Studio. Banyak compiler yang disediakan oleh Microsoft Visual Studio, misalnya saja Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe. Banyak pengembang aplikasi yang mengembangkan aplikasinya menggunakan Microsoft Visual Studio karena banyak fungsi yang telah disediakan oleh Microsoft Visual Studio, sehingga pengembang lebih mudah dan sangat terbantu dalam mengembangkan maupun membuat aplikasi [9]. Tampilan antarmuka aplikasi ini dapat dilihat pada Gambar 2.9.



Gambar 2.9 Tampilan Antarmuka Visual Studio

2.8. Blender

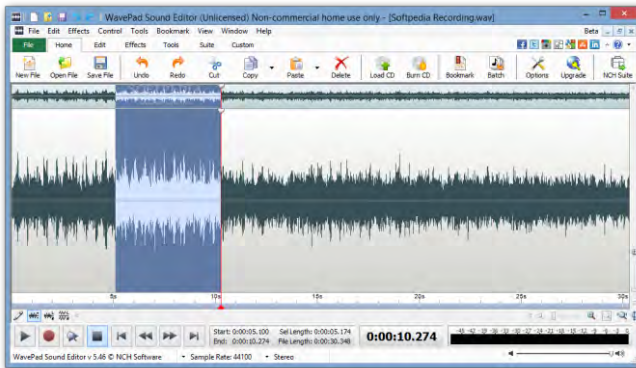
Blender adalah sebuah perangkat lunak yang berfungsi untuk permodelan 3 dimensi. Blender juga dapat digunakan untuk membuat animasi 3 dimensi. Blender tersedia dapat diunduh pada situs <https://www.blender.org/> tanpa dikenakan biaya. Blender dapat digunakan pada berbagai jenis sistem operasi. Keluaran dari Blender adalah objek-objek 3 dimensi dengan berbagai format, seperti .obj, .3ds, dan lain-lain [10]. Objek keluaran dari Blender dapat digunakan sebagai material dasar pembuatan aplikasi atau permainan dengan menggunakan *game engine* atau perangkat lunak pembuat aplikasi 3 dimensi. Tampilan antarmuka aplikasi ini dapat dilihat pada Gambar 2.10.



Gambar 2.10 Tampilan Antarmuka Blender

2.9. Wavepad Sound Editor

Software audio WavePad ini adalah software audio dengan fitur lengkap profesional dan editor musik untuk Windows. Ini memungkinkan pengguna merekam dan mengedit musik, suara dan rekaman audio lainnya. Ketika mengedit file audio, pengguna dapat memotong, menyalin dan menyisipkan bagian rekaman dan kemudian menambahkan efek seperti echo, amplifikasi dan pengurangan kebisingan. *WavePad* bekerja sebagai wav atau editor mp3, tetapi juga mendukung beberapa format file lainnya termasuk vox, gsm, wma, real audio, au, AIF, flac, ogg dan lainnya [11]. Tampilan antarmuka aplikasi ini dapat dilihat pada Gambar 2.11.



Gambar 2.11 Tampilan Wavepad Sound Editor

2.10. Bandicams

Bandicam adalah perangkat lunak yang berfungsi sebagai *screen capture*. Aplikasi ini dikembangkan oleh perusahaan software Korea Selatan Bandisoft yang dapat mengambil *screenshot* atau merekam perubahan layar. Bandicam terdiri dari dua mode. Salah satunya adalah 'area persegi pada layar' mode, yang dapat digunakan untuk merekam area tertentu di layar PC. Yang lainnya adalah 'DirectX / OpenGL window' mode, yang

dapat merekam target dibuat dalam DirectX atau OpenGL [12]. Dalam tugas akhir ini bandicams digunakan untuk merekam semua suara pada aplikasi *virtual music composer* apabila proses pembuatan lagu telah selesai. Hasil rekaman tersebut berupa sebuah video dengan format .avi dan sebuah lagu dengan format .wav. Tampilan antarmuka aplikasi ini dapat dilihat pada Gambar 2.12.



Gambar 2.12 Tampilan Antarmuka Bandicams

2.11. Tools Detection pada Aplikasi Virtual Indonesian Music Instrument (VIMi)

VIMi merupakan aplikasi yang digunakan sebagai simulasi untuk memainkan alat musik tradisional Indonesia. Terdapat beberapa alat musik daerah yang dapat di simulasikan dalam aplikasi ini, antara lain: Saron, Kenong, Angklung, dan Kendang. Pada aplikasi ini menggunakan Leap Motion Controller yang merupakan sebuah alat untuk menangkap sebuah pergerakan (*motion capture*). Pengguna cukup untuk menggunakan tangan sebagai alat *input* utama, Leap Motion akan mendeteksi koordinat

dari tangan kita untuk ditampilkan ke dalam layar monitor, pergerakan arah atas, bawah, kiri, dan kanan yang digunakan juga sesuai antara layar monitor dan objek, sedangkan untuk menekan tombol pada layar cukup untuk memajukan atau memundurkan tangan yang ditangkap pada area cakupan Leap Motion [13].

2.12. *Finger Motion* pada Aplikasi Realitas Virtual Pertunjukan Wayang Golek

Aplikasi ini merupakan aplikasi realitas virtual yang dapat mensimulasikan pertunjukan wayang golek khas tanah pasundan. Aplikasi ini dapat mempraktikkan gerakan – gerakan wayang golek yang biasa dimainkan oleh dalang. Gerakan yang dimaksud diantaranya gerakan kepala mengangguk dan gerakan tangan. Aplikasi ini dapat mempertunjukan kesenian wayang golek dalam bentuk virtual dengan sensor Leap Motion sebagai pengganti input ke dalam aplikasi. Model wayang pada aplikasi ini berupa model 3D yang dibuat menggunakan Blender. Aplikasi ini berbasis desktop yang akan berjalan di sistem operasi Windows. Dalam pembuatannya aplikasi ini dibuat menggunakan game engine Unity3D dan kode program ditulis menggunakan Bahasa C# dalam kerangka kerja .NET (menggunakan software development kit Leap Motion for Unity3D). Untuk menggerakkan model wayang, seorang dalang (orang yang memainkan wayang) menggunakan gerakan tangan dan jarinya. Gerakan tersebut akan ditangkap oleh sensor Leap Motion, lalu diterjemahkan oleh aplikasi untuk menggerakkan model wayang tersebut [14].

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan tentang analisis dan perancangan aplikasi *virtual music composer* dengan memanfaatkan teknologi Leap Motion Controller. Pembahasan yang akan dilakukan meliputi analisis fitur yang dibutuhkan dan perancangan perangkat lunak.

3.1. Analisis Perangkat Lunak

Subbab ini menjelaskan tentang hasil analisis kebutuhan perangkat lunak serta arsitektur aplikasi *virtual music composer* dengan memanfaatkan Leap Motion Controller. Tiap-tiap subbab menjelaskan tentang deskripsi umum perangkat lunak, spesifikasi kebutuhan perangkat lunak, analisis aktor, arsitektur perangkat lunak, dan skenario kasus penggunaan.

3.1.1. Deskripsi Umum Perangkat Lunak

Pada Tugas Akhir ini dilakukan pengembangan aplikasi *virtual music composer* dengan memanfaatkan teknologi Leap Motion Controller. Aplikasi ini bertujuan agar pengguna dapat mensimulasikan bermain drum dan terompet layaknya bermain alat tersebut secara nyata. Dengan memanfaatkan fitur yang dimiliki Leap Motion Controller yakni *tools and finger detection*, maka pengguna memainkan alat musik drum dan terompet menggunakan stik drum dan pergerakan jari layaknya memainkan alat tersebut secara nyata. Selanjutnya, hasil suara drum dan terompet tersebut akan dilengkapi dengan suara nyanyian oleh pengguna. Pada tahap akhir ketiga suara tersebut disatukan untuk menjadi sebuah lagu.

Terdapat tiga alat yang dimainkan pada aplikasi ini, yaitu drum, terompet, dan *microphone*. Drum yang ditampilkan pada aplikasi ini terdiri dari bass, hihat, simbal, tom-tom, snare dan *floor tom*. Dilengkapi dengan pedal untuk membunyikan bassdrum dan stik kayu untuk memainkan drum. Terompet yang dipilih dalam

aplikasi ini adalah terompet jenis bariton yakni terompet yang terdiri dari tiga tombol dan satu lubang tiup. Dalam aplikasi ini nada dasar terompet yang dimainkan adalah C. *Microphone* digunakan untuk bernyanyi. Gelombang suara yang dikeluarkan oleh pengguna akan ditangkap oleh *microphone*.

Fitur utama dalam aplikasi ini adalah rekam. Pengguna harus memainkan alat musik yang disediakan satu persatu. Pada saat akan memainkan alat musik kedua, hasil rekaman dari alat musik pertama akan dimainkan sambil pengguna memainkan alat musik kedua. Selanjutnya, pada saat pengguna akan bernyanyi, hasil rekaman dari alat musik pertama dan kedua akan dimainkan sambil pengguna bernyanyi sesuai irama dari instrument yang dihasilkan.

Terdapat beberapa tahapan agar pengguna dapat menggunakan aplikasi ini. Pertama pengguna harus memastikan bahwa komputer yang dipakai sesuai dengan spesifikasi yang dibutuhkan. Perangkat lain yang harus disediakan yaitu Leap motion Controller, stik drum, pedal drum dari *numeric pad* dan *microphone*. Dengan mekanisme yang telah diuraikan, aplikasi ini dapat difungsikan sebagai aplikasi untuk simulasi bermain drum maupun terompet sekaligus dapat menciptakan sebuah lagu dengan memanfaatkan leap motion controller.

3.1.2. Spesifikasi Kebutuhan Fungsional Perangkat Lunak

Pada sistem ini, terdapat beberapa kebutuhan fungsional yang mendukung untuk jalannya aplikasi. Fungsi yang terdapat dalam aplikasi ini adalah sebagai berikut:

- a. Memainkan suara drum
Aplikasi ini harus mampu memainkan setiap bagian dari drum jika pengguna memukulkan stik pada alat tersebut.
- b. Memainkan suara terompet
Aplikasi ini harus mampu memainkan suara terompet dengan nada C, D, E, F, G, A, B, C' dengan mendeteksi

tiupan angin dari pengguna dan pencetan pada tombol terompet yang disediakan.

- c. Menyanyikan sebuah lagu
Aplikasi ini harus mampu menerima suara dari pengguna melalui mikrofon.
- d. Merekam suara dari setiap alat
Aplikasi ini memiliki fungsional utama yaitu merekam suara dari setiap alat yang disediakan.
- e. Memainkan hasil rekaman
Aplikasi ini harus mampu memainkan hasil rekaman dari setiap alat secara bersamaan.
- f. Melihat bantuan
Aplikasi ini menyediakan bantuan yang dapat digunakan oleh pengguna apabila kesulitan dalam menggunakan aplikasi ini.

3.1.3. Identifikasi Pengguna

Dalam aplikasi ini hanya memiliki satu aktor, yaitu orang yang menggunakan aplikasi *virtual music composer* menggunakan Leap Motion Controller.

3.2. Perancangan Perangkat Lunak

Subbab ini membahas bagaimana rancangan dari aplikasi tugas akhir ini. Meliputi: Model Kasus Penggunaan, Definisi Aktor, Definisi Kasus Penggunaan, Arsitektur Umum Sistem, Rancangan Antarmuka Aplikasi, dan Rancangan Proses Aplikasi.

3.2.1. Model Kasus Penggunaan

Berdasarkan analisis spesifikasi kebutuhan fungsional dan analisis aktor dari sistem dibuat kasus penggunaan sistem. Kasus-kasus penggunaan dalam sistem ini akan dijelaskan secara rinci pada subbab ini. Kasus penggunaan digambarkan dalam sebuah

diagram kasus penggunaan. Diagram kasus penggunaan dapat dilihat pada Tabel 3.1 dan Gambar 3.1.

Tabel 3.1 Daftar Kasus penggunaan

Kode Kasus Penggunaan	Nama
UC-001	Merekam dan memainkan drum
UC-002	Merekam dan memainkan terompet
UC-003	Merekam dan menyanyikan sebuah lagu
UC-004	Memainkan hasil rekaman
UC-005	Melihat bantuan



Gambar 3.1 Diagram Kasus Penggunaan

3.2.2. Definisi Kasus Penggunaan

Detail mengenai kasus penggunaan tersebut dapat dilihat pada subbab berikut ini.

3.2.2.1. Merekam dan Memainkan Drum

Spesifikasi kasus penggunaan memainkan suara drum menggunakan Leap Motion Controller dapat dilihat pada Tabel 3.2.

Tabel 3.2 Spesifikasi Kasus Merekam dan Memainkan Drum

Nama	Merekam dan memainkan drum
Kode	UC-001
Deskripsi	Pengguna dapat merekam suara drum yang dimainkan dengan menggunakan stik drum sebagai alat pukul
Aktor	Pengguna
Kondisi Awal	Aplikasi berada pada menu awal pada <i>homescreen</i> . Pengguna sudah menyambungkan leap motion dengan aplikasi.
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>start</i>. 2. Aplikasi akan menampilkan objek drum pada monitor. 3. Pengguna memilih tombol <i>record</i> untuk memulai rekaman drum. 4. Pengguna memegang stik drum dan diarahkan di atas leap motion. Untuk selanjutnya dideteksi oleh leap motion. 5. Aplikasi akan menampilkan posisi stik pada layar monitor. Sehingga pengguna mengetahui dimana letak stik drum dan bagian mana yang akan ditabuh. 6. Pengguna memukulkan stik drum pada bagian drum yang diinginkan. 7. Pengguna memilih tombol <i>stop</i> untuk menghentikan rekaman.
- Kejadian Alternatif	A1. Pengguna menekan tombol <i>next</i> <ol style="list-style-type: none"> 1. Aplikasi akan menampilkan alat musik terompet.

	A2. Pengguna menekan tombol <i>home</i> 2. Aplikasi akan menampilkan halaman menu utama. A3. Pengguna menekan tombol <i>clear</i> 3. Aplikasi akan menghapus isi <i>list</i> rekaman drum.
Kondisi Akhir	Aplikasi akan mengeluarkan suara sesuai bagian drum yang dipukul dan <i>list</i> penyimpanan drum telah terisi.

3.2.2.2. Merekam dan Memainkan Terompet

Spesifikasi kasus penggunaan memainkan suara terompet menggunakan Leap Motion Controller dapat dilihat pada Tabel 3.3.

Tabel 3.3 Spesifikasi Kasus Merekam dan Memainkan Terompet

Nama	Merekam dan memainkan terompet
Kode	UC-002
Deskripsi	Pengguna dapat merekam suara terompet yang dimainkan dengan menggunakan mikrofon sebagai alat tiup dan tangan kiri untuk memainkan tuts terompet.
Aktor	Pengguna
Kondisi Awal	Aplikasi menampilkan alat musik drum
Aliran: - Kejadian Normal	1. Pengguna menekan tombol <i>next</i> 2. Aplikasi menampilkan objek terompet pada layar monitor 3. Pengguna memilih tombol <i>record</i> untuk memulai rekaman terompet. 4. Pengguna meniupkan angin melalui mikrofon dan menggerakkan jari tangan. 5. Aplikasi mengeluarkan bunyi terompet 6. Pengguna memilih tombol <i>stop</i> untuk menghentikan rekaman.
- Kejadian Alternatif	A1. Pengguna menekan tombol <i>next</i>

	<ol style="list-style-type: none"> 1. Aplikasi akan menampilkan alat musik mikrofon. A2. Pengguna menekan tombol <i>prev</i> <ol style="list-style-type: none"> 2. Aplikasi akan menampilkan alat musik drum. A3. Pengguna menekan tombol <i>home</i> <ol style="list-style-type: none"> 3. Aplikasi akan menampilkan halaman menu utama. A4. Pengguna menekan tombol <i>clear</i> <ol style="list-style-type: none"> 4. Aplikasi akan menghapus isi <i>list</i> rekaman terompet.
Kondisi Akhir	Aplikasi mengeluarkan bunyi nada terompet sesuai tombol yang ditekan dan <i>list</i> penyimpanan terompet telah terisi

3.2.2.3. Merekam dan Menyanyikan Sebuah Lagu

Spesifikasi kasus penggunaan menyanyikan sebuah lagu menggunakan Mikrofon dapat dilihat pada Tabel 3.4.

Tabel 3.4 Spesifikasi Kasus Penggunaan Merekam dan Menyanyikan Sebuah Lagu

Nama	Merekam dan menyanyikan sebuah lagu
Kode	UC-003
Deskripsi	Pengguna dapat merekam suara menyanyikan sebuah lagu dengan menggunakan mikrofon
Aktor	Pengguna
Kondisi Awal	Aplikasi menampilkan alat musik terompet
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>next</i>. 2. Aplikasi menampilkan objek mikrofon pada layar monitor. 3. Pengguna memilih tombol <i>record</i> untuk memulai rekaman suara vokal. 4. Pengguna menyanyikan sebuah lagu melalui mikrofon.

	5. Pengguna memilih tombol <i>stop</i> untuk menghentikan rekaman.
- Kejadian Alternatif	A1. Pengguna menekan tombol <i>prev</i> 1. Aplikasi akan menampilkan alat musik terompet. A2. Pengguna menekan tombol <i>home</i> 2. Aplikasi akan menampilkan halaman menu utama. A3. Pengguna menekan tombol <i>clear</i> 3. Aplikasi akan menghapus isi <i>list</i> rekaman mikrofon.
Kondisi Akhir	Aplikasi akan menyimpan suara yang dinyanyikan.

3.2.2.4. Memainkan Hasil Rekaman

Spesifikasi kasus penggunaan memainkan hasil rekaman dapat dilihat pada Tabel 3.5.

Tabel 3.5 Spesifikasi Kasus Penggunaan Memainkan Hasil Rekaman

Nama	Memainkan hasil rekaman
Kode	UC-004
Deskripsi	Pengguna dapat memainkan hasil rekaman secara bersamaan
Aktor	Pengguna
Kondisi Awal	Aplikasi menampilkan layar mikrofon
Aliran: - Kejadian Normal	1. Pengguna menekan tombol <i>play</i>
- Kejadian Alternatif	A1. Pengguna menekan tombol <i>home</i> 1. Aplikasi akan menampilkan layar utama

Kondisi Akhir	Aplikasi memainkan hasil rekaman yang telah disimpan sebelumnya.
----------------------	--

3.2.2.5. Melihat Bantuan

Spesifikasi kasus penggunaan melihat bantuan dapat dilihat pada Tabel 3.6.

Tabel 3.6 Spesifikasi Kasus Penggunaan Melihat Bantuan

Nama	Melihat Bantuan
Kode	UC-005
Deskripsi	Pengguna dapat melihat video tutorial sebelum memulai proses pembuatan lagu
Aktor	Pengguna
Kondisi Awal	Aplikasi menampilkan menu utama
Aliran: - Kejadian Normal	1. Pengguna menekan tombol <i>tutorial</i>
-Kejadian Alternatif	A1. Pengguna menekan tombol <i>start</i> 1. Aplikasi akan menampilkan drum. A2. Pengguna menekan tombol <i>exit</i> 2. Keluar dari aplikasi.
Kondisi Akhir	Aplikasi menampilkan video tutorial cara menggunakan aplikasi.

3.2.3. Definisi Aktor

Aktor yang terlibat dalam aplikasi *virtual music composer* dengan memanfaatkan Leap Motion Controller hanya satu yaitu pengguna. Deskripsi pengguna secara lengkap dapat dilihat pada Tabel 3.7.

Tabel 3.7 Deskripsi Pengguna

No	Nama	Deskripsi
1	Pengguna	<ul style="list-style-type: none"> - Merupakan aktor yang bertugas untuk memainkan semua alat musik. Seluruh fungsionalitas yang ada di dalam sistem berhak digunakan oleh pengguna. - Aktor harus paham terlebih dahulu mengenai jalannya aplikasi.

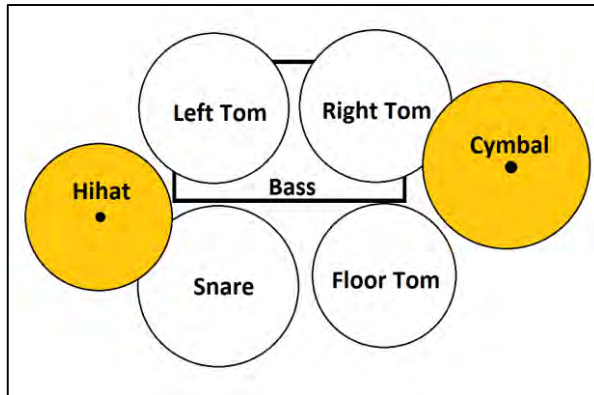
3.2.4. Perancangan Model 3D

Perancangan model diambil dari *standard asset* Unity yang dapat diakses pada <https://www.assetstore.unity3d.com/> dan selanjutnya dimodifikasi menggunakan Blender versi 2.73. Terdapat beberapa objek 3D meliputi drum, terompet, dan mikrofon. Detil rancangan model ditampilkan pada sub bab berikutnya.

3.2.4.1. Perancangan Model 3D Drum

Rancangan model 3D drum tampak pada Gambar 3.2. Perancangan model drum yang akan ditampilkan pada layar monitor adalah tampak atas. Hal ini dikarenakan menyesuaikan beberapa aplikasi virtual drum yang sudah beredar di pasaran, sebagai contoh adalah aplikasi *Real Drum – Drumset* yang dapat diunduh di *Google Play Store* [15]. Perancangan model 3D drum yang nantinya akan ditampilkan terdiri dari 7 bagian diantaranya:

1. Snare Drum
2. Left Tom
3. Right Tom
4. Floor Tom
5. Hihat
6. Cymbal
7. Bass Drum

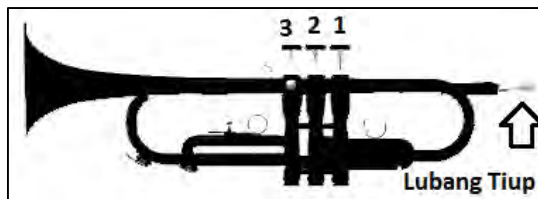


Gambar 3.2 Perancangan Model 3D Drum Tampak Atas

3.2.4.2. Perancangan Model 3D Terompet

Rancangan model 3D terompet yang akan ditampilkan merupakan terompet dengan jenis *Euphonium* atau yang sekarang lebih dikenal sebagai terompet *baritone*. Terompet *baritone* terdiri dari 3 tombol atau tuts yang dapat dipencet dengan jari serta terdapat corong untuk meniupkan angin [16]. Perancangan model terompet baritone tampak pada Gambar 3.3. Perancangan model 3D drum pada aplikasi ini nantinya terdiri dari 4 bagian penting diantaranya:

1. Tuts 1
2. Tuts 2
3. Tuts 3
4. Lubang Tiup



Gambar 3.3 Perancangan Model 3D Terompet Tampak Samping

3.2.4.3. Perancangan Model 3D Mikrofon

Pada subbab ini digambarkan rancangan model 3D mikrofon. Gambar 3.4 merupakan rancangan model 3D mikrofon yang akan ditampilkan pada sistem. Mikrofon ini memiliki bentuk yang umum digunakan pada studio rekaman.



Gambar 3.4 Perancangan Model 3D Mikrofon

3.2.5. Arsitektur Umum Sistem

Arsitektur sistem pada aplikasi *virtual music composer* dengan memanfaatkan Leap Motion Controller ini didukung oleh beberapa perangkat yaitu komputer, Leap Motion Controller, mikrofon, dan *numeric pad keyboard*. Implementasi aplikasi juga memanfaatkan salah satu aplikasi *game engine* yang sudah terkenal keunggulannya yaitu Unity. Untuk pembuatan model 3 dimensi menggunakan aplikasi *3d modeling* Blender. Arsitektur secara umum aplikasi ini terlihat pada Gambar 3.5.



Gambar 3.5 Rancangan Sederhana Arsitektur Aplikasi

3.2.6. Rancangan Antarmuka Aplikasi

Rancangan antarmuka aplikasi diperlukan untuk memberikan gambaran umum kepada pengguna bagaimana sistem yang ada dalam aplikasi ini berinteraksi dengan pengguna. Selain itu, rancangan ini juga memberikan gambaran bagi pengguna apakah tampilan yang sudah disediakan oleh aplikasi mudah untuk dipahami dan digunakan, sehingga akan muncul kesan *user experience* yang baik dan mudah.

3.2.6.1. Rancangan Antarmuka Menu Utama

Halaman ini merupakan tampilan awal dari aplikasi saat aplikasi pertama dijalankan. Pada tampilan Gambar 3.6 Rancangan Antarmuka Menu Utama terdapat tiga menu yang dapat dipilih oleh pengguna. Spesifikasi atribut antarmuka untuk halaman menu utama dapat dilihat pada Tabel 3.8.



Gambar 3.6 Rancangan Antarmuka Menu Utama

Tabel 3.8 Spesifikasi Atribut Antarmuka Menu Utama

No	Nama	Jenis	Kegunaan	Keterangan
1	Btn_Start	Button	Menampilkan antarmuka drum	
2	Btn_Tutorial	Button	Menampilkan antarmuka melihat bantuan	
3	Btn_Exit	Button	Keluar dari aplikasi	

3.2.6.2. Rancangan Antarmuka Drum

Halaman ini merupakan tampilan yang berisi objek drum. Pada tampilan Gambar 3.7 terdapat tiga tombol yang dapat dipilih oleh pengguna. Spesifikasi atribut antarmuka untuk halaman antarmuka drum dapat dilihat pada Tabel 3.9.



Gambar 3.7 Rancangan Antarmuka Drum

Tabel 3.9 Spesifikasi Atribut Antarmuka Drum

No	Nama	Jenis	Kegunaan	Keterangan
1	Btn_Right	Button	Menampilkan antarmuka terompet	
2	Btn_RecDrum Start	Button	Merekam suara drum yang dimainkan Mengaktifkan <i>Btn_RecDrum Stop</i>	
3	Btn_RecDrum Stop	Button	Menghentikan proses merekam suara drum	<i>Disable</i>
4	Btn_Home	Button	Kembali ke menu utama	
5	Btn_Clear	Button	Menghapus isi list rekaman drum	

3.2.6.3. Rancangan Antarmuka Terompet

Halaman ini merupakan tampilan yang berisi objek terompet. Pada tampilan Gambar 3.8 terdapat empat tombol yang dapat dipilih oleh pengguna. Spesifikasi atribut antarmuka untuk halaman antarmuka drum dapat dilihat pada Tabel 3.10.



Gambar 3.8 Rancangan Antarmuka Terompet

Tabel 3.10 Spesifikasi Atribut Antarmuka Terompet

No	Nama	Jenis	Kegunaan	Keterangan
1	Btn_Right	Button	Menampilkan antarmuka drum	
2	Btn_Left	Button	Menampilkan antarmuka mikrofon	
3	Btn_RecTrumpetStart	Button	Merekam suara terompet yang dimainkan Mengaktifkan <i>Btn_RecTrumpetStop</i>	
4	Btn_RecTrumpetStop	Button	Menghentikan proses	<i>Disable</i>

No	Nama	Jenis	Kegunaan	Keterangan
			merekam suara terompet	
5	Btn_Home	Button	Kembali ke menu utama	
6	Btn_Clear	Button	Menghapus isi list rekaman terompet	

3.2.6.4. Rancangan Antarmuka Mikrofon

Halaman ini merupakan tampilan yang berisi objek mikrofon. Pada tampilan Gambar 3.9 terdapat tiga tombol yang dapat dipilih oleh pengguna. Spesifikasi atribut antarmuka untuk halaman antarmuka mikrofon dapat dilihat pada Tabel 3.11.



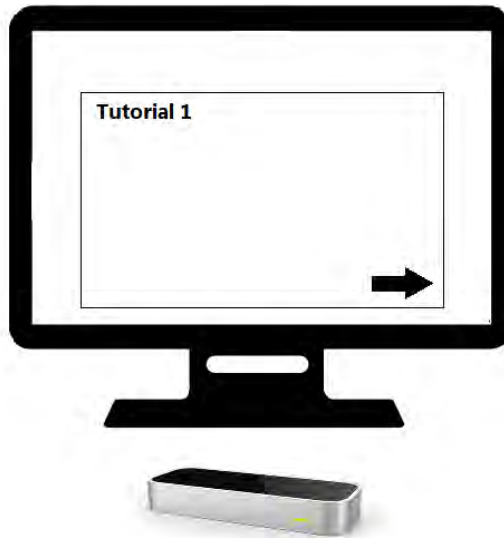
Gambar 3.9 Rancangan Antarmuka Mikrofon

Tabel 3.11 Spesifikasi Atribut Antarmuka Mikrofon

No	Nama	Jenis	Kegunaan	Keterangan
1	Btn_Left	Button	Menampilkan antarmuka terompet	
2	Btn_RecMicStart	Button	Merekam suara pengguna Mengaktifkan <i>Btn_RecMicStop</i>	
3	Btn_RecMicStop	Button	Menghentikan proses merekam suara pengguna	<i>Disable</i>
4	Btn_Home	Button	Kembali ke menu utama	
5	Btn_Clear	Button	Menghapus isi rekaman mikrofon	

3.2.6.5. Rancangan Antarmuka Melihat Bantuan

Halaman ini merupakan tampilan yang berisi video tutorial. Pada tampilan Gambar 3.10 Rancangan Antarmuka Melihat Bantuan terdapat tiga tombol yang dapat dipilih oleh pengguna. Spesifikasi atribut antarmuka untuk halaman antarmuka melihat bantuan dapat dilihat pada Tabel 3.12.



Gambar 3.10 Rancangan Antarmuka Melihat Bantuan

Tabel 3.12 Spesifikasi Atribut Antarmuka Melihat Bantuan

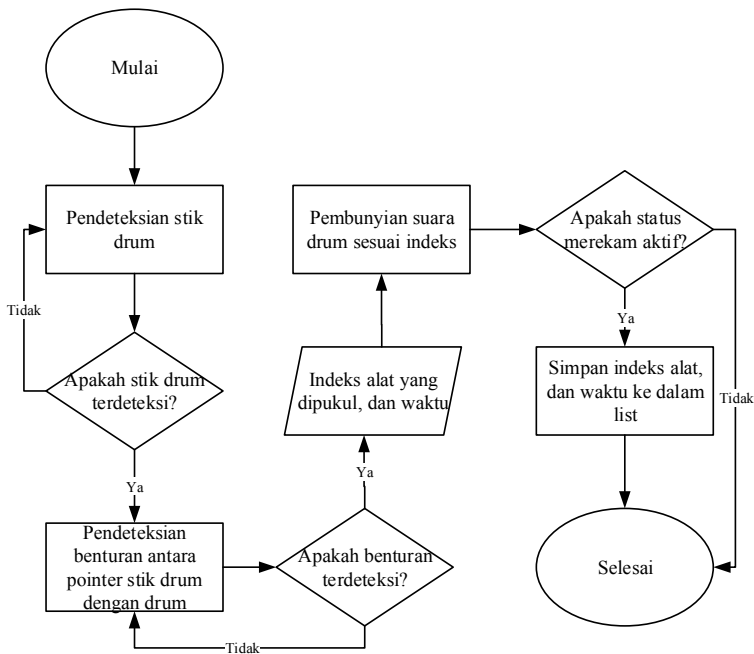
No	Nama	Jenis	Kegunaan	Keterangan
1	Btn_Next	Button	Melanjutkan ke tutorial selanjutnya	

3.2.7. Perancangan Proses Aplikasi

Pada subbab ini akan dijelaskan mengenai rancangan proses yang dilakukan untuk mendukung fungsionalitas yang sudah dirancang pada aplikasi. Rancangan ini diperlukan untuk memetakan proses yang ada mulai dari awal hingga akhir. Proses-proses terdiri dari proses memainkan suara drum, proses memainkan suara terompet, proses menyanyikan sebuah lagu, proses memainkan hasil rekaman, dan proses melihat bantuan.

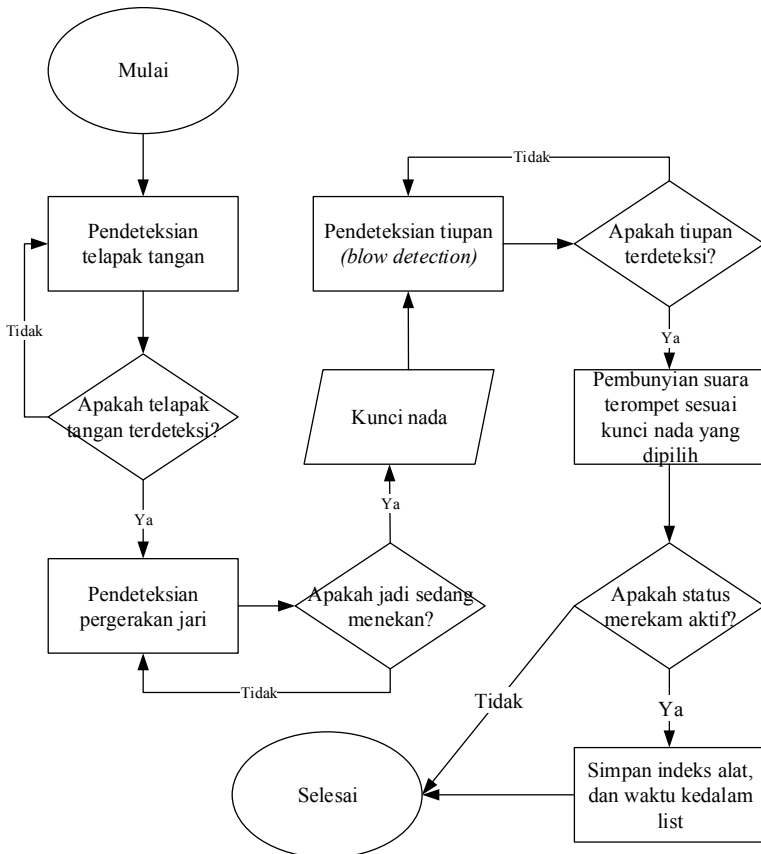
3.2.7.1. Rancangan Proses Memainkan Drum

Dalam rancangan proses memainkan drum terdapat beberapa tahapan yang harus dibuat. Tahapan pertama adalah pendeteksian stik drum. Tahapan kedua adalah pendeteksian benturan, suara drum akan berbunyi apabila terjadi benturan atau tabrakan antara pointer stik drum pada layar dengan drum itu sendiri. Tahapan ketiga yaitu proses penyimpanan kedalam *list* apabila pengguna mengaktifkan tombol *record*. Setiap bagian pada drum sudah diberi indeks untuk membedakan masing-masing bagian tersebut. Apabila tombol *record* aktif maka indeks alat yang dipukul dan waktu kapan alat tersebut dipukul akan disimpan kedalam *list*. *Flowchart* rancangan proses memainkan drum ditunjukkan pada Gambar 3.11.



Gambar 3.11 Flowchart Rancangan Proses Memainkan Drum

3.2.7.2. Rancangan Proses Memainkan Terompet



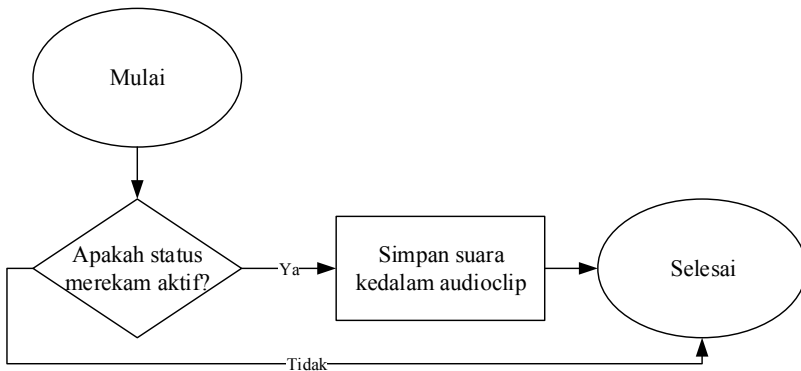
Gambar 3.12 Flowchart Rancangan Proses Memainkan Terompet

Dalam rancangan proses memainkan terompet terdapat beberapa tahapan yang harus dibuat. Tahapan pertama adalah pendeteksian telapak tangan. Tahapan kedua adalah pendeteksian pergerakan jari, suara terompet akan berbunyi apabila jari sedang menekan. Tahapan ketiga yaitu proses pendeteksian tiupan (*blow detection*) dengan cara menganalisa tiupan yang diterima oleh

mikrofon. Tahapan keempat adalah proses penyimpanan kedalam *list* apabila pengguna mengaktifkan tombol *record*. Nada yang dimainkan nantinya akan disimpan kedalam *list* beserta dengan waktu saat nada tersebut dimainkan. *Flowchart* rancangan proses memainkan drum ditunjukkan pada Gambar 3.12.

3.2.7.3. Rancangan Proses Menyanyikan Sebuah Lagu

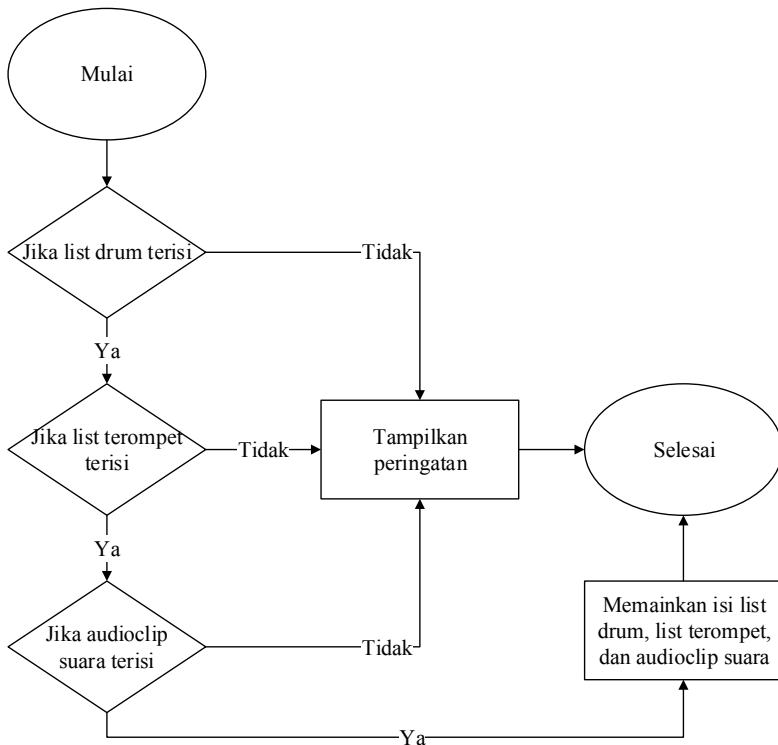
Dalam rancangan proses menyanyikan sebuah lagu hanya terdapat satu tahapan saja yang harus dibuat, tahapan tersebut adalah dengan menyimpan suara nyanyian pengguna kedalam audioclip apabila tombol *record* sedang aktif. Proses ini tidak dapat dijalankan apabila pengguna tidak menekan tombol *record* terlebih dahulu. *Flowchart* rancangan proses menyanyikan sebuah lagu dapat dilihat pada Gambar 3.13.



Gambar 3.13 Flowchart Rancangan Proses Menyanyikan Sebuah lagu

3.2.7.4. Rancangan Proses Memainkan Hasil Rekaman

Dalam rancangan proses memainkan hasil rekaman terlebih dahulu dilakukan pengecekan terhadap isi dari *list drum*, *list terompet*, dan *audioclip* suara nyanyian pengguna. Apabila ketiga atribut tersebut belum terisi maka sistem akan mengeluarkan peringatan. Namun apabila ketiga atribut tersebut tidak kosong maka akan dimainkan secara bersamaan dan menghasilkan sebuah lagu. *Flowchart* rancangan proses memainkan hasil rekaman ditunjukkan pada Gambar 3.14.



Gambar 3.14 Flowchart Rancangan Proses Memainkan Hasil Rekaman

3.2.7.5. Rancangan Proses Melihat Bantuan

Dalam rancangan proses melihat bantuan akan ditampilkan beberapa perintah-perintah yang harus dilakukan oleh pengguna. Perintah pertama berisi pengguna harus memastikan semua alat sudah terpasang dengan komputer. Perintah kedua, pengguna mencoba memposisikan stik drum berada di atas *Leap Motion* lalu menggerakkan keatas dan kebawah layaknya sedang memukul drum. Perintah ketiga, pengguna harus dapat membunyikan bass drum dengan cara menginjak pedal drum. Perintah keempat, pengguna harus meniupkan angin melalui mikrofon untuk dapat membunyikan terompet. Perintah terakhir, pengguna mencoba bernyanyi menggunakan mikrofon.

BAB IV IMPLEMENTASI

Bab ini akan menjelaskan tentang implementasi Tugas Akhir berdasarkan rancangan perangkat lunak. Proses implementasi mengacu pada rancangan perangkat yang telah dilakukan sebelumnya, namun juga dimungkinkan terjadinya perubahan-perubahan jika dirasa perlu. Implementasi dilakukan dalam bahasa C#.

4.1. Lingkungan Implementasi

Sub bab ini menjelaskan tentang lingkungan implementasi perangkat lunak yang dibangun. Lingkungan selama proses implementasi aplikasi *virtual music composer* dengan menggunakan Leap Motion Controller adalah sebagai berikut:

Tabel 4.1 Spesifikasi Lingkungan Implementasi

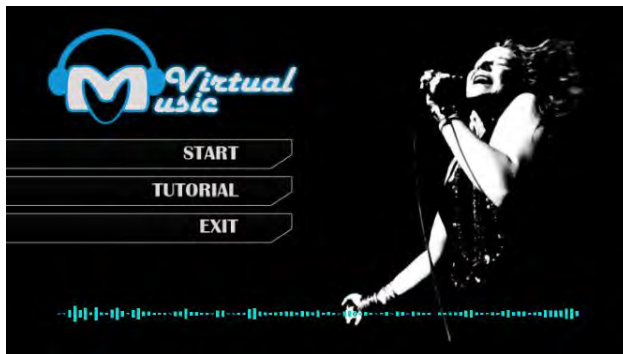
Perangkat Keras	<ul style="list-style-type: none">- Prosesor Intel(R) Core(TM) i3-2120 CPU @ 3.30GHz- Memori 4 GB- Monitor- <i>Numeric Keyboard</i>- <i>Speaker</i>- Mikrofon- Leap motion Controller
Perangkat Lunak	<ul style="list-style-type: none">- Sistem Operasi Microsoft Windows 8 Pro 64-bit- Blender 2.7.3- Unity 5.2.1f1- Visual Studio 2013- Leap Motion SDK- Photoshop CS5- Bendicam

4.2. Implementasi Antar muka

Subbab ini akan menjelaskan tentang implementasi dari antar muka yang digunakan. Antarmuka aplikasi ini terdiri dari lima tampilan.

4.2.1. Impelementasi Antarmuka Menu Utama

Tampilan pertama merupakan menu utama. Pada tampilan ini terdapat tiga tombol yaitu “*start*” yang berfungsi untuk memulai proses membuat lagu. Tombol kedua merupakan “*tutorial*” yang berfungsi untuk menampilkan video yang berisi tutorial menggunakan aplikasi. Dan tombol yang ketiga yaitu “*exit*” untuk keluar dari program. Tampilan antarmuka pada menu utama dapat dilihat pada Gambar 4.1.

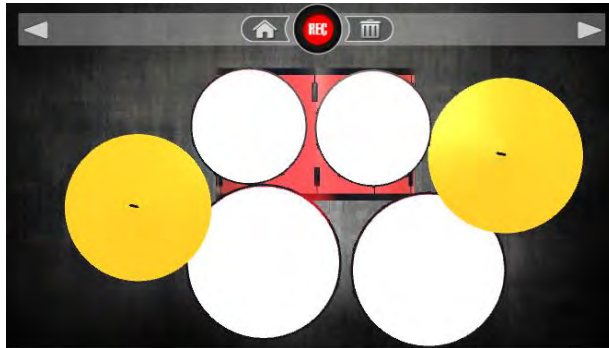


Gambar 4.1 Implementasi Antarmuka Menu Utama

4.2.2. Impelementasi Antarmuka Drum

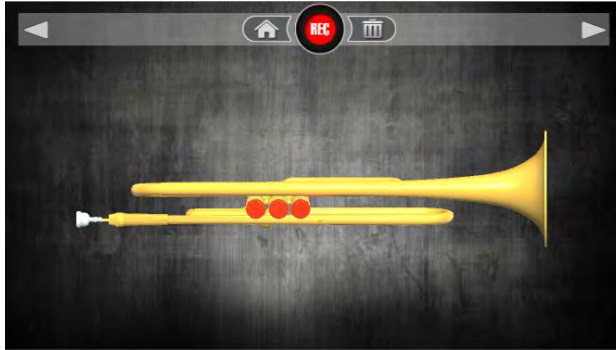
Tampilan kedua merupakan tampilan drum. Pada tampilan ini terdapat tiga tombol, pertama adalah tombol “*close*” berfungsi untuk mengembalikan pada tampilan menu utama. Tombol kedua adalah “*record*” yang berfungsi untuk merekam setiap suara yang

dihasilkan dari permainan drum oleh pengguna. Tombol ketiga adalah “*next*” yang berfungsi untuk menampilkan layar terompet. Tampilan antarmuka drum dapat dilihat pada Gambar 4.2.



Gambar 4.2 Implementasi Antarmuka Drum

4.2.3. Impelementasi Antarmuka Terompet



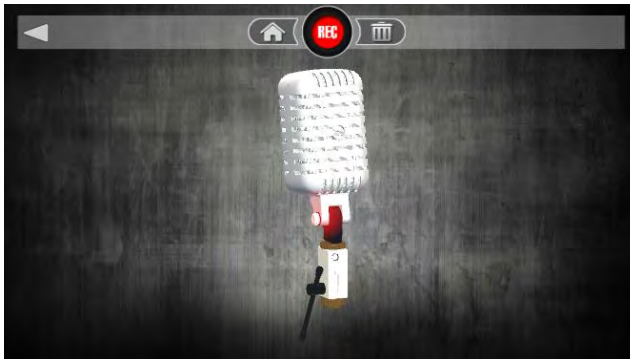
Gambar 4.3 Implemntasi Antarmuka Terompet

Tampilan ketiga merupakan tampilan terompet. Pada tampilan ini terdapat empat tombol, pertama adalah tombol “*close*” berfungsi untuk mengembalikan pada tampilan menu utama. Tombol kedua adalah “*record*” yang berfungsi untuk merekam setiap suara yang dihasilkan dari permainan terompet oleh

pengguna. Tombol ketiga adalah “*next*” yang berfungsi untuk menampilkan layar mikrofon. Tombol keempat adalah “*prev*” yang berfungsi untuk menampilkan layar drum. Tampilan antarmuka terompet dapat dilihat pada Gambar 4.3.

4.2.4. Implementasi Antarmuka Mikrofon

Tampilan keempat merupakan tampilan mikrofon. Pada tampilan ini terdapat tiga tombol, pertama adalah tombol “*close*” berfungsi untuk mengembalikan pada tampilan menu utama. Tombol kedua adalah “*record*” yang berfungsi untuk merekam setiap suara yang dihasilkan dari permainan terompet oleh pengguna. Tombol ketiga adalah “*prev*” yang berfungsi untuk menampilkan layar drum. Tampilan antarmuka mikrofon dapat dilihat pada Gambar 4.4.



Gambar 4.4 Implementasi Antarmuka Mikrofon

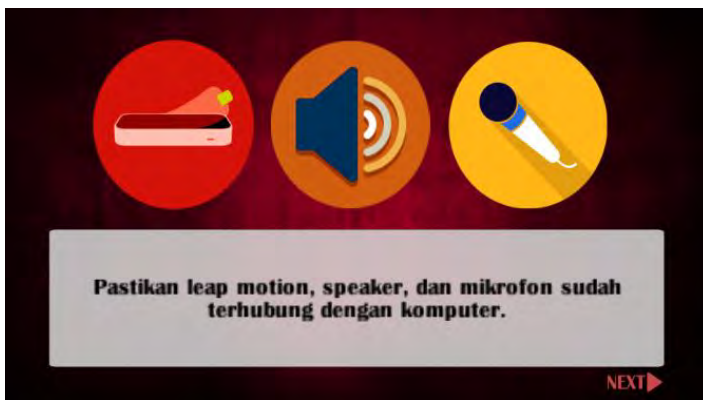
4.2.5. Implementasi Antarmuka Melihat Bantuan

Tampilan ini merupakan tampilan melihat bantuan. Pada tampilan ini terdapat perubahan dari tahap perencanaan. Bantuan atau tutorial yang diberikan bukan berupa video melainkan tahapan-tahapan yang harus dilalui oleh pengguna. Pengguna akan

melihat tutorial pertama hingga terakhir dengan menekan satu tombol yang disediakan yaitu tombol *next*. Tahapan yang harus dilalui pengguna pada saat melihat bantuan adalah sebagai berikut:

4.2.5.1. Memastikan Hardware Sudah Terpasang

Pengguna harus memastikan hardware yang digunakan sudah terpasang atau terhubung dengan komputer. Hardware yang dimaksud pada tahapan ini adalah leap motion controller, speaker, dan mikrofon. Apabila pengguna sudah menghubungkan ketiga hardware tersebut pengguna dapat menekan tombol *next* untuk melihat tahapan bantuan berikutnya. Tampilan antarmuka untuk melihat bantuan tahap pertama dapat dilihat pada Gambar 4.5.

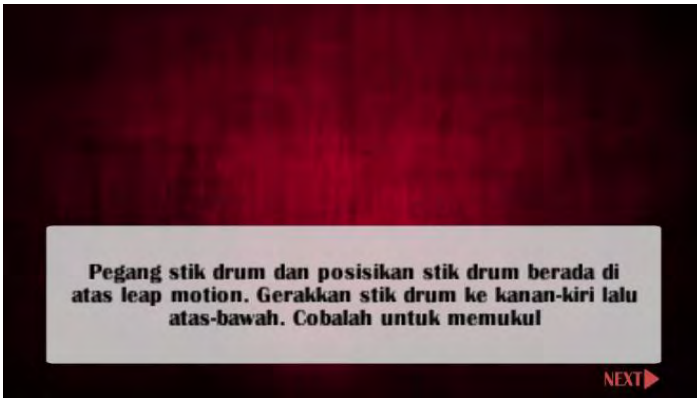


Gambar 4.5 Implementasi Antarmuka Melihat Bantuan Tahap Pertama

4.2.5.2. Mencoba Deteksi Stik Drum

Pengguna mencoba deteksi stik drum oleh leap motion controller dengan memegang kedua stik tepat diatas leap motion controller. Pengguna penggerakkan stik drum keatas dan kebawah lalu ke kanan dan kekiri agar pengguna dapat mengenali deteksi

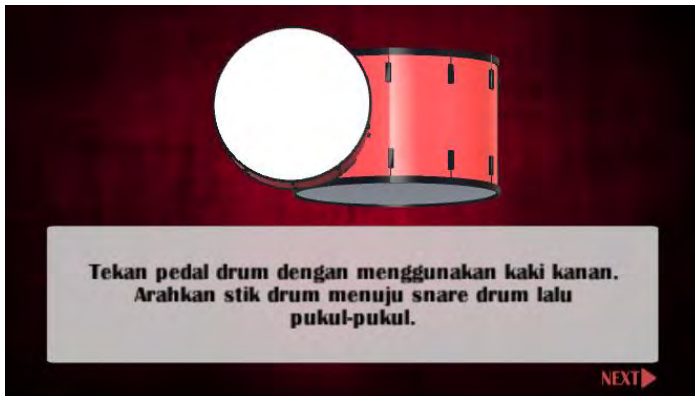
yang dilakukan oleh leap motion controller. Apabila pengguna sudah menggerakkan stik drum maka pengguna dapat menekan tombol *next* untuk melihat tahapan bantuan berikutnya. Tampilan antarmuka untuk melihat bantuan tahap kedua dapat dilihat pada Gambar 4.6.



Gambar 4.6 Implementasi Antarmuka Melihat Bantuan Tahap Kedua

4.2.5.3. Mencoba Bermain Drum

Pengguna mencoba bermain drum sederhana yang terdiri dari bass drum dan snare drum saja. Hal ini bertujuan agar pengguna dapat mengetahui cara bagaimana membunyikan drum. Pengguna dapat membunyikan bass drum dengan menginjak pedal drum yang disediakan. Pengguna juga dapat membunyikan snare drum dengan memukulkan stik drum pada snare drum. Apabila pengguna sudah dapat membunyikan drum sederhana tersebut maka pengguna dapat menekan tombol *next* untuk melihat tahapan bantuan berikutnya. Tampilan antarmuka untuk melihat bantuan tahap ketiga dapat dilihat pada Gambar 4.7.



Gambar 4.7 Implementasi Antarmuka Melihat Bantuan Tahap Ketiga

4.2.5.4. Mencoba meniup Terompet

Pengguna mencoba membunyikan terompet dengan meniupkan angin ke arah mikrofon. Apabila pengguna sudah dapat membunyikan terompet maka pengguna dapat menekan tombol *next* untuk melihat tahapan bantuan berikutnya. Tampilan antarmuka untuk melihat bantuan tahap keempat dapat dilihat pada Gambar 4.8.

4.2.5.5. Mencoba Bernyanyi

Pengguna mencoba bernyanyi menggunakan mikrofon. Apabila pengguna sudah dapat bernyanyi dengan menggunakan mikrofon maka pengguna dapat menekan tombol *next* untuk melihat tahapan bantuan berikutnya. Tampilan antarmuka untuk melihat bantuan tahap kelima dapat dilihat pada Gambar 4.9.



Gambar 4.8 Implementasi Antarmuka Melihat Bantuan Tahap Keempat



Gambar 4.9 Implementasi Antarmuka Melihat Bantuan Tahap Kelima

4.3. Implementasi Proses

Sub bab ini akan menjelaskan tentang implementasi dari rancangan proses yang dijelaskan pada Bab III. Penjelasan mengenai implementasi proses ini dibagi berdasarkan komponen-

komponen aplikasi. Berikut ini merupakan penjelasan dari tiap-tiap komponen.

4.3.1. Implementasi Melihat Bantuan

Proses ini merupakan proses dimana pengguna melihat video yang berisi arahan untuk menggunakan aplikasi ini secara lengkap. Implementasi proses melihat bantuan menggunakan bantuan fungsi dari Unity yaitu “Movie Texture”. Langkah-langkah implementasi proses melihat bantuan adalah pertama pengguna memilih tombol “*Tutorial*” pada menu utama. Lalu pengguna memainkan video yang disediakan. Implementasi video tutorial untuk melihat bantuan dapat dilihat pada potongan Kode Sumber 4.1.

```
using UnityEngine;
using System.Collections;
using UnityEngine.UI;

public class Tutorial : MonoBehaviour {

    private int indexTutorial;
    public Text instructionText;
    public GameObject drum3d, terompet3d, mic3d, icon;
    public AudioSource audioTerompet, audioBassDrum,
audioSnareDrum;

    // ===== ATTRIBUTE BLOW DETECTION
    private int sampleCount = 1024;
    private float refValue = 0.1f;
    private float threshold = 0.02f;
    private float rmsValue;
    private float dbValue;
    public float pitchValue;
    private float[] samples;
    private float[] spectrum;
    private bool boolean;
    private AudioSource audioCanvas;

    // Use this for initialization
    void Start () {
        instructionText.text = "Pastikan leap motion,
speaker, dan mikrofon sudah terhubung dengan komputer.";
        indexTutorial = 1;
    }
}
```

```

// ===== START -> BLOW DETECTION
if (Microphone.devices.Length <= 0)
Debug.LogWarning("Microphone not connected!");
else
{
    samples = new float[sampleCount];
    spectrum = new float[sampleCount];
    audioCanvas = GetComponent();
    audioCanvas.clip = Microphone.Start("Built-in
Microphone", false, 999, 48000);
    audioCanvas.Play();
}
}

// Update is called once per frame
void Update () {
    if (indexTutorial == 2)
    {
        icon.SetActive(false);
        instructionText.text = "Pegang stik drum dan
posisi stik drum berada di atas leap motion. Gerakkan
stik drum ke kanan-kiri lalu atas-bawah. Cobalah untuk
memukul";
    }
    if (indexTutorial == 3)
    {
        drum3d.SetActive(true);
        instructionText.text = "Tekan pedal drum
dengan menggunakan kaki kanan. Arahkan stik drum menuju
snare drum lalu pukul-pukul.";
        if (Input.GetKeyDown(KeyCode.Keypad3))
        {
            audioBassDrum.Play();
        }
    }
    if (indexTutorial == 4)
    {
        drum3d.SetActive(false);
        terompet3d.SetActive(true);
        instructionText.text = "Tiupkan angin melalui
mikrofon.";
        if (Microphone.devices.Length <= 0)
Debug.LogWarning("Microphone not connected!");
        else
        {
            AnalyzeSound();
            if (rmsValue > 0.66 && !boolean)
            {

```

```

        audioTerompet.Play();
        boolean = true;
    }
    else if (rmsValue < 0.1 && boolean) // rms
Value Rata2
    {
        boolean = false;
    }
}
}
if (indexTutorial == 5)
{
    terompet3d.SetActive(false);
    mic3d.SetActive(true);
    instructionText.text = "Bernyanyilah
menggunakan mikrofon.";
}
if (indexTutorial == 6)
{
    mic3d.SetActive(false);
    instructionText.text = "Selesai...";
}
if (indexTutorial == 7)
{
    Application.LoadLevel(0);
}
}

// Mendeteksi triget pukulan dan pencetan
void OnTriggerEnter(Collider collision)
{
    if (collision.gameObject.name == "Snare")
    {
        audioSnareDrum.Play();
    }
}

// next button
public void NextTutorial() {
    indexTutorial += 1;
}

// Mendeteksi atribut tiupan
void AnalyzeSound()
{
    audioCanvas.GetOutputData(samples, 0); // Get all
of our samples from the mic.

    // Sums squared samples

```

```

float sum = 0;
for (int i = 0; i < sampleCount; i++)
{
    sum += Mathf.Pow(samples[i], 2);
}

rmsValue = Mathf.Sqrt(sum / sampleCount);
dbValue = 20 * Mathf.Log10(rmsValue / refValue);

if (dbValue < -160)
{
    dbValue = -160;
}

// Gets the sound spectrum.
audioCanvas.GetSpectrumData(spectrum, 0,
FFTWindow.BlackmanHarris);
float maxV = 0;
int maxN = 0;

// Find the highest sample.
for (int i = 0; i < sampleCount; i++)
{
    if (spectrum[i] > maxV && spectrum[i] >
threshold)
    {
        maxV = spectrum[i];
        maxN = i; // maxN is the index of max
    }
}

// Pass the index to a float variable
float freqN = maxN;

// Interpolate index using neighbours
if (maxN > 0 && maxN < sampleCount - 1)
{
    float dL = spectrum[maxN - 1] /
spectrum[maxN];
    float dR = spectrum[maxN + 1] /
spectrum[maxN];
    freqN += 0.5f * (dR * dR - dL * dL);
}
}

```

Kode Sumber 4.1 Melihat Bantuan

4.3.2. Implementasi Memainkan Drum

Proses ini merupakan proses pertama yang harus dilakukan oleh pengguna. Pengguna memainkan drum dengan menggunakan stik drum yang telah dideteksi lalu memposisikan stik di atas leap motion dan memukulkannya kearah bawah serta menginjakkan kaki kanan pada pedal yang disediakan untuk membunyikan bass drum. Pengguna dapat merekam suara drum yang sedang dimainkan dengan memilih tombol “*record*” yang berada pada sisi atas layar Apabila pengguna sudah selesai memainkan drum, pengguna memilih tombol “*stop recording*”.

4.3.2.1. Mendeteksi Stik Drum

Aplikasi harus dapat mendeteksi alat yang digunakan untuk memainkan drum yaitu stik drum. Stik drum dapat dideteksi menggunakan *class* yang disediakan oleh *Leap Motion SDK* yaitu *ToolModel*. Didalam *class* ini terdapat dua fungsi utama yaitu *CreateTool* yang berfungsi untuk membuat suatu objek yang merepresentasikan benda yang dideteksi, dan *UpdateToolModel* yang berfungsi untuk memperbarui posisi dari benda yang dideteksi. Potongan kode untuk mendeteksi stik dapat dilihat pada Kode Sumber 4.2.

```
/** Creates a ToolModel instance. */
protected ToolModel CreateTool(ToolModel model)
{
    ToolModel tool_model = Instantiate(model,
transform.position, transform.rotation) as ToolModel;
    tool_model.gameObject.SetActive(true);
    Leap.Utils.IgnoreCollisions(tool_model.gameObject,
gameObject);
    return tool_model;
}

protected void UpdateToolModels(Dictionary<int, ToolModel>
all_tools, ToolList leap_tools, ToolModel model)
{
    List<int> ids_to_check = new List<int>(all_tools.Keys);
```

```

// Go through all the active tools and update them.
int num_tools = leap_tools.Count;
for (int h = 0; h < num_tools; ++h) {
    Tool leap_tool = leap_tools[h];

    // Only create or update if the tool is enabled.
    if (model) {

        ids_to_check.Remove(leap_tool.Id);

        // Create the tool and initialized it if it doesn't
exist yet.
        if (!all_tools.ContainsKey(leap_tool.Id)) {
            ToolModel new_tool = CreateTool(model);
            new_tool.SetController(this);
            new_tool.SetLeapTool(leap_tool);
            new_tool.InitTool();
            all_tools[leap_tool.Id] = new_tool;
        }

        // Make sure we update the Leap Tool reference.
        ToolModel tool_model = all_tools[leap_tool.Id];
        tool_model.SetLeapTool(leap_tool);
        tool_model.MirrorZAxis(mirrorZAxis);

        // Set scaling.
        tool_model.transform.localScale = new Vector3(30.0f,
30.0f, 30.0f);

        tool_model.UpdateTool();
    }
}

// Destroy all tools with defunct IDs.
for (int i = 0; i < ids_to_check.Count; ++i) {
    Destroy(all_tools[ids_to_check[i]].gameObject);
    all_tools.Remove(ids_to_check[i]);
}
}

```

Kode Sumber 4.2 Mendeteksi Stik

4.3.2.2. Mendeteksi Pukulan Stik Drum

Fungsi *OnTriggerEnter* merupakan fungsi pendeteksian benturan atau tabrakan antar objek. Objek yang dimaksud pada fungsi ini adalah benturan antara stik drum dan alat drum tersebut. Apabila terjadi benturan maka aplikasi akan mengeluarkan suara dari alat tersebut. Potongan kode untuk memainkan drum dijelaskan pada Kode Sumber 4.3.

```
void OnTriggerEnter(Collider collision)
{
    if (collision.gameObject.name == "HiHat")
        i = 0;
    else if (collision.gameObject.name == "LeftTom")
        i = 1;
    else if (collision.gameObject.name == "RightTom")
        i = 2;
    else if (collision.gameObject.name == "Cymbal")
        i = 3;
    else if (collision.gameObject.name == "Snare")
        i = 4;
    else if (collision.gameObject.name == "FloorTom")
        i = 5;

    VoiceDrumPlay(i);

    // Save to list if button record clicked
    if (R.isDrumRecord == 1)
    {
        R.SaveToListDrum(i);
    }
}

// Membunyikan suara drum pada audiosource
public void VoiceDrumPlay(int i)
{
    audioDrum[i].clip = audioclip[i];
    audioDrum[i].Play();
}
```

Kode Sumber 4.3 Memainkan Drum

4.3.2.3. Merekam Suara Drum

Implementasi merekam suara drum menggunakan metode penyimpanan pada *list* dua dimensi. Variabel yang disimpan pada *list* yaitu index alat drum dan waktu kapan alat tersebut dibunyikan. Potongan kode untuk merekam suara drum dijelaskan pada Kode Sumber 4.4.

```
// Tombol StartRecordDrum
public void StartDrumRecord()
{
    recButton.SetActive(false);
    stopButton.SetActive(true);
    rightButton.SetActive(false);
    listDrumRecSave.Clear();
    isDrumRecord = true;
    startDrumRecTime = Time.time;
}

// Tombol StopRecordDrum
public void StopDrumRecord()
{
    stopButton.SetActive(false);
    recButton.SetActive(true);
    rightButton.SetActive(true);
    stopDrumRecTime = Time.time;
    isDrumRecord = false;
    isDrumFill = true;
}

// Memasukkan waktu dan alat yang ditabuh
public void SaveToListDrum(int index_suara)
{
    listDrumRecSave.Add(new Vector2(index_suara, Time.time -
    startDrumRecTime));
}
```

Kode Sumber 4.4 Merekam Suara Drum

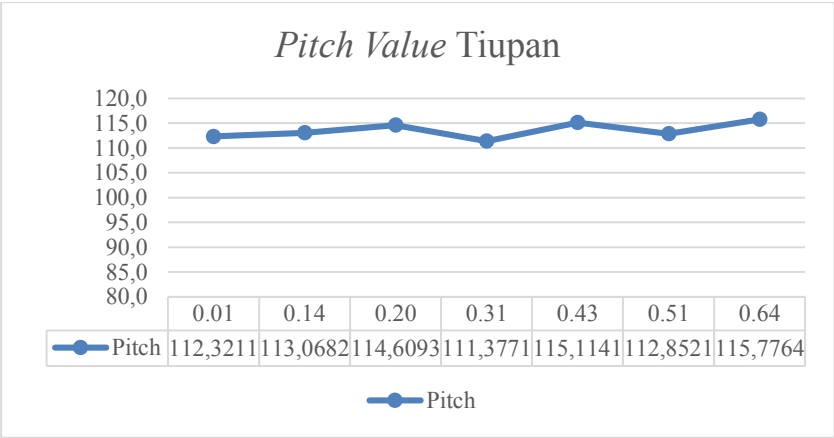
4.3.3. Implementasi Memainkan Suara Terompet

Proses ini merupakan proses kedua yang harus dilakukan oleh pengguna. Pengguna memainkan terompet dengan meniupkan angin melalui mikrofon sambil menekan tuts terompet dengan

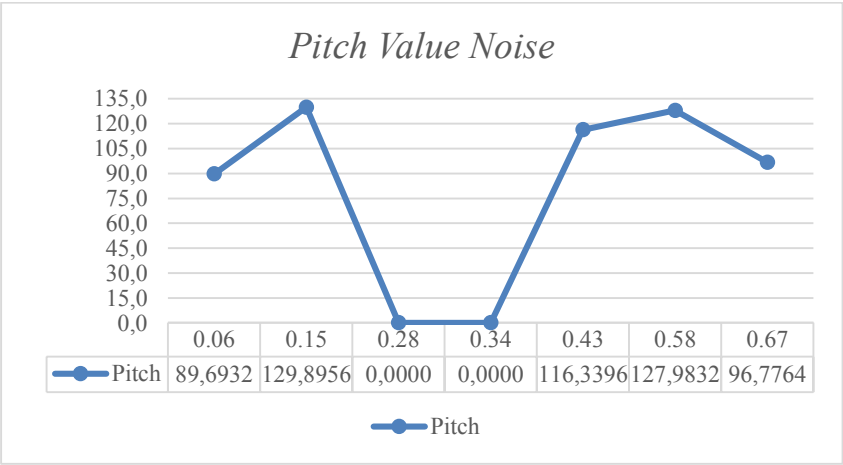
memposisikan jari telunjuk, jari tengah dan jari manis di atas leap motion dan menggerakkannya keatas dan kebawah. Pengguna dapat merekam suara terompet yang dimainkan dengan memilih tombol “*record*” yang berada pada sisi atas layar. Pengguna akan mendengarkan hasil rekaman dari drum yang telah disimpan sebelumnya jika pengguna menekan tombol “*record*”. Apabila pengguna sudah selesai memainkan terompet, pengguna dapat memilih tombol “*stop recording*”.

4.3.3.1. Mendeteksi Tiupan

Fungsi *GetSpectrumData* merupakan fungsi yang digunakan untuk mendeteksi tiupan dengan menganalisis suara yang diterima oleh mikrofon. Dalam hal ini suara yang dimaksud adalah angin yang ditiupkan oleh pengguna melalui mikrofon. Analisis dilakukan untuk mendapatkan nilai *pitch*. Dikatakan pengguna sedang meniup apabila nilai *pitch* yang didapat lebih dari 110 dan nilai tersebut bertahan selama 0.5 detik. Nilai 110 didapat dari batas minimum nilai *pitch* yang didapat dari tiupan pengguna. Sedangkan nilai 0.5 detik didapat dari waktu minimum *pitch* bernilai stabil yaitu diatas 110. Hal ini bertujuan untuk membedakan *noise* yang diterima mikrofon dari lingkungan dengan *blow* atau tiupan dari pengguna. Perbedaan nilai *pitch* yang didapat dari tiupan dan *noise* lingkungan dapat dilihat pada Gambar 4.10 dan Gambar 4.11. Potongan kode untuk mendeteksi tiupan dijelaskan pada Kode Sumber 4.5.



Gambar 4.10 Grafik Pitch Value Dari Tiupan Pengguna



Gambar 4.11 Grafik Pitch Value Dari Noise Lingkungan

```
Void Update()  
{  
    if (Microphone.devices.Length <= 0)  
        Debug.LogWarning ("Microphone not connected!");  
    else {  
        AnalyzeSound ();  
        if (pitchValue > 110 && !blow) {
```

```

        blow = true;
        blowSeconds = blowStatic;
    }
    if (pitchValue == 0 && blow) {
        blowSeconds -= Time.deltaTime;
        if (blowSeconds < 0) {
            blow = false;
        }
    }
}

// Mendeteksi tiupan
void AnalyzeSound()
{
    audio.GetOutputData(samples, 0);
    float sum = 0;
    for (int i = 0; i < sampleCount; i++)
    {
        sum += Mathf.Pow(samples[i], 2);
    }
    rmsValue = Mathf.Sqrt(sum / sampleCount);
    dbValue = 20 * Mathf.Log10(rmsValue / refValue);

    // Clamp it to -160dB min
    if (dbValue < -160)
    {
        dbValue = -160;
    }

    // Gets the sound spectrum.
    audio.GetSpectrumData(spectrum, 0,
        FFTWindow.BlackmanHarris);
    float maxV = 0;
    int maxN = 0;

    // Find the highest sample.
    for (int i = 0; i < sampleCount; i++)
    {
        if (spectrum[i] > maxV && spectrum[i] > threshold)
        {
            maxV = spectrum[i];
            maxN = i; // maxN is the index of max
        }
    }

    float freqN = maxN;
    if (maxN > 0 && maxN < sampleCount - 1)
    {

```

```

        float dL = spectrum[maxN - 1] / spectrum[maxN];
        float dR = spectrum[maxN + 1] / spectrum[maxN];
        freqN += 0.5f * (dR * dR - dL * dL);
    }
    // Convert index to frequency
    pitchValue = freqN * 24000 / sampleCount;
}

```

Kode Sumber 4.5 Mendeteksi Tiupan

4.3.3.2. Mendeteksi Telapak Tangan

Aplikasi harus dapat mendeteksi telapak tangan yang nantinya jari telunjuk, jari tengah, dan jari manis digunakan untuk memainkan tuts terompet. Jari-jari tangan dapat dideteksi menggunakan *class* yang disediakan oleh *Leap Motion SDK* yaitu *HandModel*. Didalam *class* ini terdapat dua fungsi utama yaitu *CreateHand* yang berfungsi untuk membuat suatu objek yang merepresentasikan telapak tangan yang sedang dideteksi, dan *UpdateHandModel* yang berfungsi untuk memperbarui setiap perubahan dari telapak tangan yang dideteksi. Potongan kode untuk mendeteksi telapak tangan dapat dilihat pada Kode Sumber 4.6.

```

protected HandModel CreateHand(Hand leap_hand, HandModel
model)
{
    HandModel hand_model = Instantiate(model,
transform.position, transform.rotation)
        as HandModel;
    hand_model.gameObject.SetActive(true);
    Leap.Utils.IgnoreCollisions(hand_model.gameObject,
gameObject);
    hand_model.transform.SetParent(transform);
    hand_model.SetLeapHand(leap_hand);
    hand_model.MirrorZAxis(mirrorZAxis);
    hand_model.SetController(this);

    handEvent handHandler = onCreateHand;
    if (handHandler != null) {
        handHandler(hand_model);
    }

    return hand_model;
}

```

```

}

protected void UpdateHandModels(Dictionary<int, HandModel>
all_hands, HandList leap_hands, HandModel left_model,
HandModel right_model)
{
    List<int> ids_to_check = new List<int>(all_hands.Keys);

    // Go through all the active hands and update them.
    int num_hands = leap_hands.Count;
    for (int h = 0; h < num_hands; ++h) {
        Hand leap_hand = leap_hands[h];

        HandModel model = (mirrorZAxis != leap_hand.IsLeft) ?
left_model : right_model;

        // If we've mirrored since this hand was updated,
        destroy it.
        if (all_hands.ContainsKey(leap_hand.Id) &&
            all_hands[leap_hand.Id].IsMirrored() !=
mirrorZAxis) {
            DestroyHand(all_hands[leap_hand.Id]);
            all_hands.Remove(leap_hand.Id);
        }

        // Only create or update if the hand is enabled.
        if (model != null) {
            ids_to_check.Remove(leap_hand.Id);

            // Create the hand and initialized it if it doesn't
            exist yet.
            if (!all_hands.ContainsKey(leap_hand.Id)) {
                HandModel new_hand = CreateHand(leap_hand, model);

                // Set scaling based on reference hand.
                float hand_scale = MM_TO_M * leap_hand.PalmWidth /
new_hand.handModelPalmWidth;
                new_hand.transform.localScale = hand_scale *
Vector3.one;

                new_hand.InitHand();
                new_hand.UpdateHand();
                all_hands[leap_hand.Id] = new_hand;
            } else {
                // Make sure we update the Leap Hand reference.
                HandModel hand_model = all_hands[leap_hand.Id];
                hand_model.SetLeapHand(leap_hand);
                hand_model.MirrorZAxis(mirrorZAxis);
            }
        }
    }
}

```

```

        // Set scaling based on reference hand.
        float hand_scale = MM_TO_M * leap_hand.PalmWidth /
hand_model.handModelPalmWidth;
        hand_model.transform.localScale = hand_scale *
Vector3.one;
        hand_model.UpdateHand();
    }
}

// Destroy all hands with defunct IDs.
for (int i = 0; i < ids_to_check.Count; ++i) {
    DestroyHand(all_hands[ids_to_check[i]]);
    all_hands.Remove(ids_to_check[i]);
}
}

```

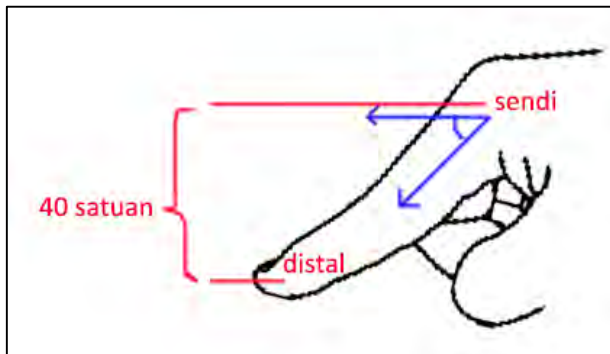
Kode Sumber 4.6 Mendeteksi Telapak Tangan

4.3.3.3. Mendeteksi Jari Yang Sedang Menekan

Implementasi dari proses ini dengan membandingkan posisi sendi *metacarpophalangeal* dengan ruas tulang *distal* pada jari telunjuk, jari tengah, dan jari manis. Sendi dan ruas tulang tersebut dapat dilihat pada Gambar 4.12. Apabila posisi ruas tulang distal lebih rendah 40 satuan daripada posisi sendi *metacarpophalangeal* pada sumbu *y* maka dapat disimpulkan bahwa jari tersebut sedang menekan. Nilai 40 satuan didapat melalui observasi bahwa disaat jari membentuk sudut 45° kearah bawah, selisih posisi sendi dengan ruas tulang distal adalah 40 satuan, untuk lebih detil dapat dilihat pada Gambar 4.13. Potongan kode untuk memainkan terompet dijelaskan pada Kode Sumber 4.7.



Gambar 4.12 Gambar Tulang dan Sendi Pada Jari



Gambar 4.13 Penentuan Selisih Nilai Y

```
void FixedUpdate() {
    if (sendiTelunjuk.transform.position.y -
        jariTelunjuk.transform.position.y >= 40.0f) {
        tutl = true;
    } else {
        tutl = false;
    }
}
```

```

    }
    if (sendiTengah.transform.position.y -
jariTengah.transform.position.y >= 40.0f) {
        tut2 = true;
    } else {
        tut2 = false;
    }
    if (sendiManis.transform.position.y -
jariManis.transform.position.y >= 40.0f) {
        tut3 = true;
    } else {
        tut3 = false;
    }

    if (tut1)
        i = 1;
    if (tut2)
        i = 2;
    if (tut3)
        i = 3;
    if (tut1 && tut2)
        i = 4;
    if (tut1 && tut3)
        i = 5;
    if (tut2 && tut3)
        i = 6;
    if (tut1 && tut2 && tut3)
        i = 7;
    if (!tut1 && !tut2 && !tut3)
        i = 0;

    SetI (i);

    if (onTrigger) {
        VoiceStop ();
        onTrigger = false;
        boolean = false;
    }
    if (B.blow && !boolean) {
        VoiceStop ();
        VoiceTrumpetPlay (i);
        if (R.isTrumpetRecord)
            R.SaveToListTrumpet (i);
        onTrigger = false;
        boolean = true;
    }
    if (!B.blow) {
        VoiceStop ();
        boolean = false;
    }

```



```

    }

}

// Membunyikan suara trumpet pada audiosource1
public void VoiceTrumpetPlay(int i)
{
    audiosource1.clip = audioclip[i];
    audiosource1.Play();
}

// Menghentikan suara trumpet jika tidak sedang meniup
public void VoiceStop()
{
    audiosource1.Stop();
}

// Mengatur perubahan pada kunci nada
private void SetI(int i) {
    if (i != iPlay) {
        iLast = iPlay;
        iPlay = i;
        onTrigger = true;
    }
}
}

```

Kode Sumber 4.7 Mendeteksi Jari yang Sedang Menekan

4.3.3.4. Merekam Suara Terompet

Implementasi merekam suara terompet menggunakan metode penyimpanan pada *list* dua dimensi. Variabel yang disimpan pada *list* yaitu index not terompet dan waktu kapan not tersebut dibunyikan. Potongan kode untuk merekam suara terompet dijelaskan pada Kode Sumber 4.3.

```

// Memasukkan waktu dan alat yang ditiup
public void SaveToListTrumpet(int index_suara)
{
    listTrumpetRecSave.Add(new Vector2(index_suara,
    Time.time - startTrumpetRecTime));
}

```

Kode Sumber 4.8 Merekam Suara Terompet

4.3.4. Implementasi Menyanyikan Sebuah Lagu

Proses ini merupakan proses ketiga yang harus dilakukan oleh pengguna. Aplikasi akan menampilkan objek mikrofon. Pengguna memilih tombol “*record*” yang berada pada sisi atas layar. Pengguna akan mendengarkan hasil rekaman dari drum dan terompet yang telah disimpan sebelumnya. Disaat yang bersamaan pengguna dapat menyanyikan lagu sesuai iringan musik. Apabila pengguna sudah selesai menyanyikan sebuah lagu, pengguna memilih tombol “*stop recording*”.

Implementasi dari proses ini dengan menggunakan fungsi yang sudah disediakan oleh Unity yaitu *MicrophoneStart*. Fungsi *MicrophoneStart* merupakan fungsi untuk menerima inputan suara dari mikrofon. Potongan kode untuk menyanyikan sebuah lagu dijelaskan pada Kode Sumber 4.2.

```
void Start () {
    if (Microphone.devices.Length <= 0) print("Microphone
not connected!");
    else
    {
        micConnected = true;
        Microphone.GetDeviceCaps(null, out minFreq, out
maxFreq);
        if (minFreq == 0 && maxFreq == 0) maxFreq = 44100;
    }
}
public void StartMicrophoneRecord() {
    if (micConnected)
    {
        if (!Microphone.IsRecording(null))
        {
            goAudioSource.clip = Microphone.Start(null,
true, 100, maxFreq);
        }
    } else
    {
        warningText.GetComponent<Text>().text = "Mikrofon
tidak tersambung";
        warningPanel.SetActive(true);
    }
}
```

Kode Sumber 4.9 Menyanyikan Sebuah Lagu

4.3.5. Implementasi Memainkan Hasil Rekaman

Tahapan ini merupakan proses terakhir dimana aplikasi harus mampu memainkan hasil rekaman yang telah disimpan secara bersamaan. Potongan kode untuk memainkan hasil rekaman dijelaskan pada Kode Sumber 4.10.

```
void FixedUpdate()
{
    if (isDrumPlayed) {
        if (listDrumRecSave[a].y <= Time.time -
stopDrumRecTime) {
            int idx = (int)listDrumRecSave[a].x;
            VoiceDrumPlay(idx);
            a += 1;
        }
    }
    if (isTrumpetPlayed) {
        if (listTrumpetRecSave[b].y <= Time.time -
stopTrumpetRecTime) {
            int idx = (int)listTrumpetRecSave[b].x;
            VoiceTrumpetPlay(idx);
            b += 1;
        }
    }
    if (isMicPlayed) {
        goAudioSource.Play();
        isMicPlayed = false;
    }

    if (a == listDrumRecSave.Count) isDrumPlayed = false;
    if (b == listTrumpetRecSave.Count) isTrumpetPlayed =
false;
}

public void PlayAllRec()
{
    a = 0;
    b = 0;
    isTrumpetPlayed = 1;
    isDrumPlayed = 1;
    isMicPlayed = 1;
    stopDrumRecTime = stopMicRecTime;
    stopTrumpetRecTime = stopMicRecTime;
}
```

Kode Sumber 4.10 Memainkan Hasil Rekaman

4.4. Implementasi Integrasi Leap Motion dengan Unity

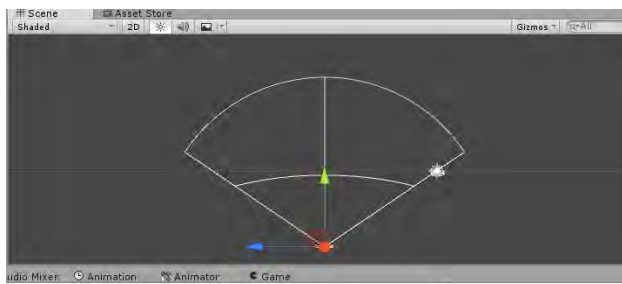
Implementasi integrasi Leap Motion dengan Unity memerlukan suatu penghubung, dalam hal ini adalah SDK Leap Motion. SDK Leap Motion dapat diunduh melalui banyak situs. Salah satu situs yang menyediakan SDK Leap Motion adalah situs resmi Leap Motion yaitu <https://developer.leapmotion.com/>. Hasil unduh dari SDK ini berupa sebuah *package* untuk mengintegrasikan Leap Motion dengan Unity. *Package* tersebut kemudian diimpor kedalam Unity. Setelah proses impor berhasil, maka akan muncul empat buah folder baru pada bagian *Assets* di Unity. Hasil proses tersebut dapat dilihat pada Gambar 4.14.

Empat folder baru tersebut adalah folder Leap Motion, Leap Motion VR, dan *Plugins*. Folder Leap Motion berisikan asset-asset yang digunakan untuk membuat aplikasi atau game dengan memanfaatkan Leap Motion. Folder leap Motion VR berisikan asset-asset untuk membangun aplikasi *virtual reality* sedangkan folder *Plugins* berisi pengaturan yang ada pada Leap Motion.

Untuk implementasi sensor area Leap Motion, maka digunakan *Hand Detection Scene* yang ada di dalam folder Leap Motion. *Scene* tersebut dimasukan ke dalam *scene* Unity. Hasil proses tersebut dapat dilihat pada Gambar 4.15.



Gambar 4.14 Hasil Proses Impor *Package* Leap Motion



Gambar 4.15 Implementasi Input Area Leap Motion

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada aplikasi yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsionalitas sistem dan kegunaan sistem. Pengujian fungsionalitas mengacu pada kasus penggunaan pada bab tiga. Pengujian kegunaan program dilakukan dengan mengetahui tanggapan dari pengguna terhadap sistem. Hasil evaluasi menjabarkan tentang rangkuman hasil pengujian pada bagian akhir bab ini.

5.1. Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan Tugas Akhir ini dilakukan pada lingkungan dan alat kakas seperti yang tertera pada Tabel 5.1.

Tabel 5.1 Lingkungan Pengujian Sistem

Perangkat Keras	<ul style="list-style-type: none">- Prosesor Intel(R) Core(TM) i3-2120 CPU @ 3.30GHz- Memori 4 GB- Monitor- <i>Numeric Keyboard</i>- <i>Speaker</i>- Mikrofon- Leap motion Controller
Perangkat Lunak	<ul style="list-style-type: none">- Sistem Operasi Microsoft Windows 8 Pro 64-bit- Blender 2.7.3- Unity 5.2.1f1- Visual Studio 2013- Leap Motion SDK- Photoshop CS5- Bendicam

5.2. Pengujian Aplikasi

Pengujian aplikasi dilakukan untuk mengetahui kesesuaian keluaran dari tiap tahap atau langkah penggunaan fitur terhadap skenario yang dipersiapkan. Berikut ini penjabaran skenario dan hasil uji coba yang dilakukan terhadap perangkat lunak yang dibangun.

5.2.1. Skenario Pengujian Fungsionalitas

Pada subbab ini dijelaskan beberapa skenario uji coba perangkat lunak secara mandiri berdasarkan metode kotak hitam sebagai dasar tolak ukur keberhasilan. Pengujian fungsionalitas yang terdapat pada aplikasi dijabarkan sebagai berikut:

- a. Uji coba merekam dan memainkan drum
- b. Uji coba merekam dan memainkan terompet
- c. Uji coba merekam dan menyanyikan sebuah lagu
- d. Uji coba memainkan hasil rekaman
- e. Uji coba melihat bantuan

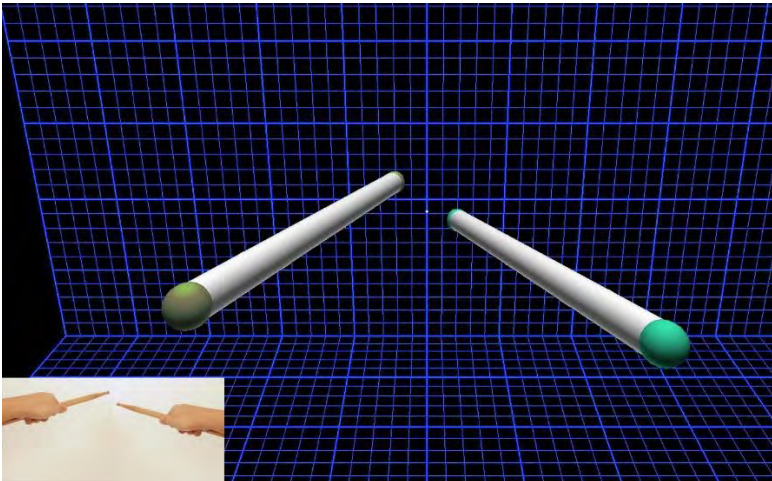
Daftar uji coba tersebut merupakan pengujian yang dilakukan untuk menguji setiap kasus penggunaan pada perangkat lunak yang dibangun. Dibuat beberapa skenario yang dilakukan pada setiap daftar pengujian tersebut. Penjelasan mengenai cara dan hasil pengujian fungsionalitas perangkat lunak dibahas pada subbab hasil uji coba.

5.2.2. Hasil Uji Coba Fungsionalitas

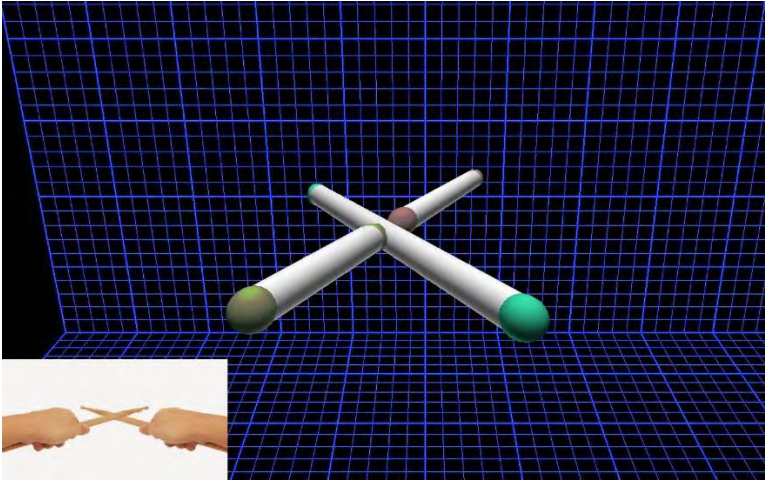
Pada sub bab ini dijelaskan secara detail mengenai skenario yang dilakukan dan hasil yang didapatkan dari pengujian fungsionalitas perangkat lunak yang dibangun. Penjelasan disajikan dengan menampilkan kondisi awal, masukan, keluaran, hasil yang dicapai, dan kondisi akhir. Berikut ini merupakan penjabaran skenario dan hasil pengujian yang dicapai pada tiap-tiap fungsionalitas perangkat lunak.

5.2.2.1. Uji Coba Merekam dan Memainkan Drum

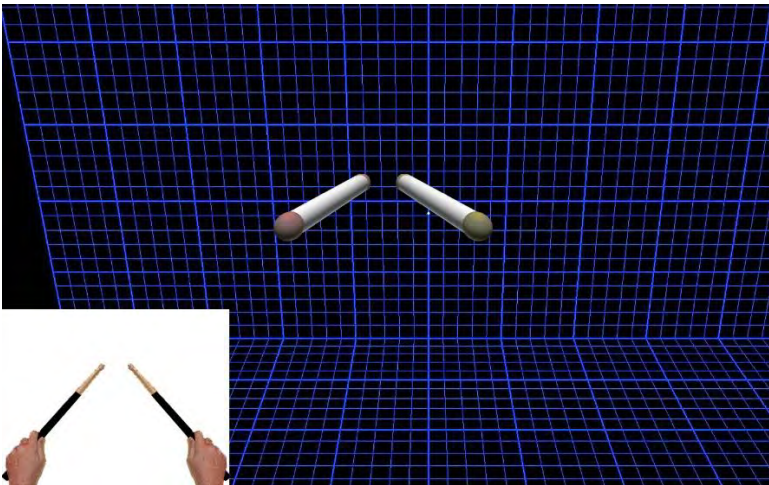
Pada ujicoba ini dilakukan pengujian terhadap benda yang digunakan untuk mengetahui apakah pendeteksian stik drum dapat dilakukan secara sempurna atau tidak. Gambar 5.1 menunjukkan stik drum dapat terdeteksi. Gambar 5.2 menunjukkan stik drum dalam keadaan menyilang dengan posisi stik kanan berada di bawah stik kiri. Hasil pendeteksian stik drum tersebut memperlihatkan bahwa stik drum yang berada di atas yaitu stik kiri terpotong menjadi 2 bagian. Hal ini menunjukkan leap motion tidak dapat mendeteksi stik drum dalam keadaan menyilang. Dilakukan percobaan yang lain dengan melapisi stik drum menggunakan solasi hitam pada bagian pangkal. Hal ini bertujuan untuk memperpendek stik yang terdeteksi oleh leap motion. Gambar 5.3 menunjukkan stik yang sudah diberi solasi hitam dalam keadaan tidak menyilang. Gambar 5.4 menunjukkan stik yang sudah diberi solasi hitam dalam keadaan menyilang.



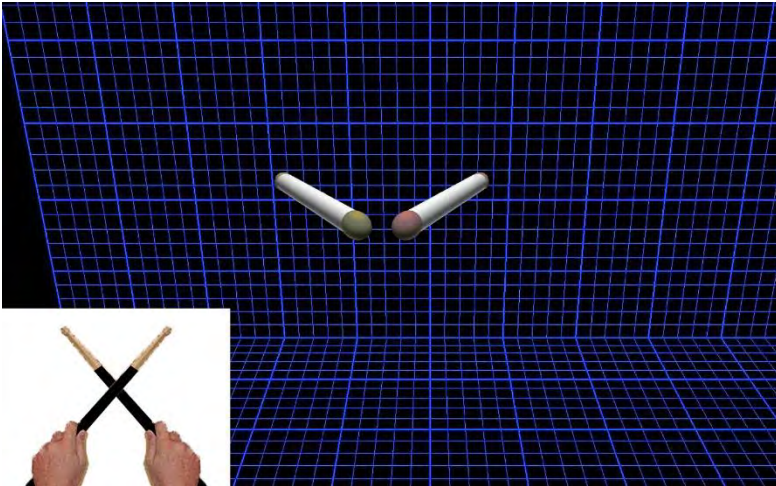
Gambar 5.1 Stik Biasa Dalam Keadaan Tidak Menyilang



Gambar 5.2 Stik Biasa Dalam Keadaan Menyilang



Gambar 5.3 Stik Diberi Isolasi Dalam Posisi Tidak Menyilang



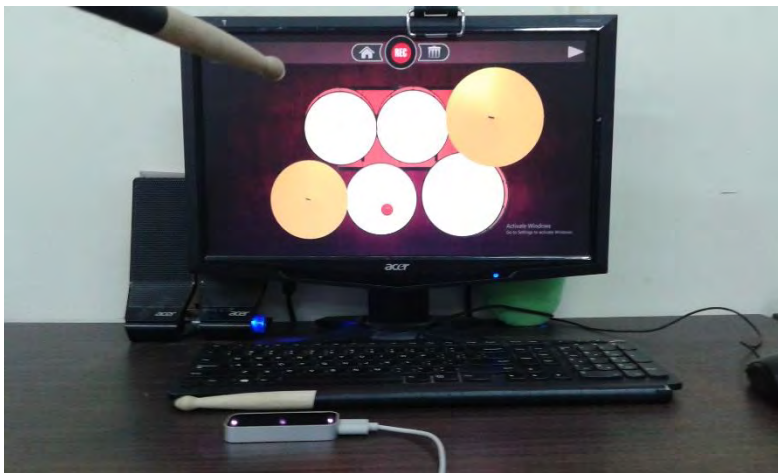
Gambar 5.4 Stik Diberi Isolasi Dalam Posisi Menyilang

Uji coba memainkan drum ini berfungsi untuk mengetahui keberhasilan aplikasi dalam menjalankan skenario merekam dan memainkan drum. Hasil dari pengujian tersebut secara lengkap dapat dilihat pada Tabel 5.2. Hasil dari skenario uji coba tersebut dapat dilihat pada Gambar 5.5 dan Gambar 5.6.

Tabel 5.2. Hasil Uji Coba Merekam dan Memainkan Drum

ID	UJ-P-01
Nama	Uji Coba Merekam dan Memainkan Drum
Tujuan Uji Coba	Pengguna menyelesaikan skenario merekam dan memainkan drum
Kondisi awal	Pengguna sudah memilih tombol “ <i>start</i> ” untuk memulai dan pengguna harus memastikan alat-alat tambahan sudah terpasang, seperti: speaker, leap motion, mikrofon, dan pedal
Skenario 1	Pengguna memilih tombol <i>record</i> lalu pengguna memainkan setiap alat drum

	dengan memijakkan kaki kanan diatas pedal dan memukulkan stik drum kearah leap motion. Pengguna memilih tombol <i>stop record</i> apabila telah selesai memainkan drum.
Keluaran yang diharapkan	Sistem akan mengeluarkan suara sesuai dengan alat yang dipukul dan <i>list</i> penyimpanan drum terisi sesuai dengan waktu dan alat yang dipukul
Hasil uji coba	Berhasil
Kondisi akhir	Sistem berhasil mengeluarkan suara sesuai dengan alat yang dipukul. <i>List</i> berhasil menyimpan waktu dan <i>index</i> alat yang dipukul



Gambar 5.5 Uji Coba Saat Memainkan Drum



Gambar 5.6 Uji Coba Saat Merekam Drum

5.2.2.2. Uji Coba Merekam dan Memainkan Terompet

Uji coba memainkan terompet ini berfungsi untuk mengetahui keberhasilan aplikasi dalam menjalankan scenario merekam dan memainkan terompet. Hasil dari pengujian tersebut secara lengkap dapat dilihat pada Tabel 5.3. Hasil dari skenario uji coba tersebut dapat dilihat pada Gambar 5.7 dan Gambar 5.8.

Tabel 5.3 Hasil Uji Coba Merekam dan Memainkan Terompet

ID	UJ-P-02
Nama	Uji Coba Merekam dan Memainkan Terompet
Tujuan Uji Coba	Pengguna menyelesaikan skenario merekam dan memainkan terompet
Kondisi awal	Pengguna sudah memilih tombol “start” untuk memulai dan pengguna harus memastikan alat-alat tambahan sudah terpasang, seperti: speaker, leap motion, mikrofon, dan pedal

Skenario 1	Pengguna memilih tombol <i>record</i> lalu memainkan setiap tuts terompet dengan meniupkan angin melalui mikrofon dan menekan tuts terompet dengan memposisikan telapak tangan kiri diatas leap motion lalu menggerakkan jari telunjuk, jari tengah dan jari manis Pengguna memilih tombol <i>stop record</i> apabila telah selesai memainkan terompet.
Keluaran yang diharapkan	Sistem akan mengeluarkan suara sesuai dengan tuts yang dipencet, <i>list</i> penyimpanan berisi waktu dan nada terompet yang dibunyikan
Hasil uji coba	Berhasil
Kondisi akhir	Sistem berhasil mengeluarkan suara sesuai dengan tuts yang dipencet dan <i>list</i> berhasil menyimpan waktu serta nada yang dibunyikan



Gambar 5.7 Uji Coba Saat Memainkan Terompet



Gambar 5.8 Uji Coba Saat Merekam Terompet

5.2.2.3. Uji Coba Merekam dan Menyanyikan Sebuah Lagu

Uji coba merekam dan menyanyikan sebuah lagu ini berfungsi untuk mengetahui keberhasilan aplikasi dalam menjalankan scenario merekam dan menyanyikan sebuah lagu. Hasil dari pengujian tersebut secara lengkap dapat dilihat pada Tabel 5.4. Hasil dari skenario uji coba tersebut dapat dilihat pada Gambar 5.9 dan Gambar 5.10.

Tabel 5.4 Uji Coba Menyanyikan Sebuah Lagu

ID	UJ-P-03
Nama	Uji Coba Merekam dan Menyanyikan Sebuah Lagu.
Tujuan Uji Coba	Pengguna menyelesaikan skenario merekam dan menyanyikan sebuah lagu.

Kondisi awal	Pengguna sudah memilih tombol “ <i>start</i> ” untuk memulai dan pengguna harus memastikan alat-alat tambahan sudah terpasang, seperti: speaker, leap motion, mikrofon, dan pedal.
Skenario 1	Pengguna memilih tombol <i>record</i> lalu menyanyikan sebuah lagu melalui mikrofon. Pengguna memilih tombol <i>stop record</i> apabila telah selesai menyanyikan sebuah lagu.
Keluaran yang diharapkan	Sistem akan menyimpan suara yang masuk melalui mikrofon.
Hasil uji coba	Berhasil.
Kondisi akhir	Sistem berhasil menyimpan suara yang masuk melalui mikrofon kedalam <i>audioclip</i> .



Gambar 5.9 Uji Coba Saat Menyanyikan Sebuah Lagu



Gambar 5.10 Uji Coba Saat Merekam Mikrofon

5.2.2.4. Uji Coba Memainkan Hasil Rekaman

Uji coba merekam setiap alat berfungsi untuk mengetahui keberhasilan aplikasi dalam menjalankan skenario memainkan hasil rekaman secara bersamaan. Hasil dari pengujian tersebut secara lengkap dapat dilihat pada Tabel 5.5.

Tabel 5.5 Uji Coba Merekam Setiap Alat

ID	UJ-P-04
Nama	Uji Coba Merekam Setiap Alat.
Tujuan Uji Coba	Pengguna menyelesaikan skenario memainkan hasil rekaman.
Kondisi awal	Pengguna sudah menyelesaikan scenario pada UJ-P-01, UJ-P-02, dan UJ-P-03.
Skenario 1	Pengguna menekan tombol <i>save and play</i> untuk memainkan hasil rekaman secara bersamaan.

Keluaran yang diharapkan	Aplikasi dapat memainkan lagu hasil dari rekaman secara bersamaan
Hasil uji coba	Berhasil.
Kondisi akhir	Sistem berhasil memainkan lagu hasil dari rekaman secara bersamaan.

5.2.2.5. Uji Coba Melihat Bantuan

Uji coba melihat bantuan ini berfungsi untuk mengetahui keberhasilan aplikasi dalam menjalankan skenario pengguna melihat bantuan. Hasil dari pengujian tersebut secara lengkap dapat dilihat pada Tabel 5.6.

Tabel 5.6 Uji Coba Melihat Bantuan

ID	UJ-P-05
Nama	Uji Coba Melihat Bantuan.
Tujuan Uji Coba	Pengguna menyelesaikan skenario melihat bantuan.
Kondisi awal	Pengguna sudah memilih tombol “ <i>tutorial</i> ” untuk dapat melihat video tutorial yang disediakan.
Skenario 1	Pengguna menekan tombol “ <i>start</i> ” untuk memulai video Pengguna menekan tombol “ <i>stop</i> ” untuk menghentikan video
Keluaran yang diharapkan	Sistem akan memutar video tutorial.
Hasil uji coba	Berhasil.
Kondisi akhir	Sistem berhasil menampilkan <i>step by step</i> tutorial.

5.3. Pengujian Pengguna

Pengujian pada perangkat lunak yang dibangun tidak hanya dilakukan pada fungsionalitas yang dimiliki, tetapi juga ditujukan kepada pengguna untuk mencoba secara langsung. Pengujian ini berfungsi sebagai pengujian subjektif yang bertujuan untuk mengetahui tingkat keberhasilan aplikasi yang dibangun dari sisi pengguna. Hal ini dapat dicapai dengan meminta penilaian dan tanggapan dari pengguna terhadap sejumlah aspek perangkat lunak yang ada.

5.3.1. Skenario Pengujian Pengguna

Dalam melakukan pengujian perangkat lunak, pengguna diminta mencoba menggunakan perangkat lunak untuk mencoba semua fungsionalitas dan fitur yang ada. Pengujian aplikasi oleh pengguna dilakukan dengan sebelumnya memberikan informasi seputar aplikasi, kegunaan, dan fitur-fitur yang dimiliki. Setelah informasi tersampaikan, pengguna kemudian diarahkan untuk langsung mencoba aplikasi dengan spesifikasi lingkungan yang sama dengan yang telah diuraikan pada Lingkungan Pengujian.

Jumlah pengguna yang terlibat dalam pengujian perangkat lunak sebanyak delapan orang. Dalam melakukan pengujian, pengguna melakukan percobaan lebih dari satu kali penggunaan untuk masing-masing pengguna.

Dalam memberikan penilaian dan tanggapan, pengguna diberikan kuesioner pengujian perangkat lunak. Kuesioner pengujian perangkat lunak ini memiliki beberapa aspek penilaian seputar desain antarmuka, *immersivity*, dan tingkat kenyamanan aplikasi. Nilai yang diberikan memiliki rentang nilai 1 hingga 6 dengan rincian pada Tabel 5.7. Pada bagian akhir terdapat saran untuk perbaikan fitur. Detail kuesioner pengguna dapat dilihat pada Tabel 5.8.

Tabel 5.7 Rentang Nilai

Nilai	Keterangan
1	Sangat Tidak Setuju (STS)
2	Tidak Setuju (TS)
3	Kurang Setuju (KS)
4	Cukup Setuju (CS)
5	Setuju (S)
6	Sangat Setuju (SS)

Tabel 5.8 Kuesioner Pengguna

No	Parameter Antarmuka	Nilai (1-6)
1	Aplikasi memiliki tampilan, warna dan desain yang menarik	
2	Aplikasi memiliki tata letak <i>button</i> yang mudah dilihat/dikenali	
3	Aplikasi memiliki objek dan background yang sesuai dengan tema music	
No	Parameter Immersivity	Nilai (1-6)
4	Saya merasakan sensasi nyata seperti bermain drum sungguhan	
5	Saya merasakan sensasi nyata seperti bermain terompet sungguhan	
6	Saya merasakan sensasi nyata seperti bernyanyi sungguhan	
No	Parameter Kenyamanan	Nilai (1-6)
7	Aplikasi dapat berjalan lancar tanpa adanya <i>lag</i> dan <i>crash</i>	
8	Saya merasa terbantu dengan adanya tutorial yang disediakan	
9	Saya merasa nyaman selama menggunakan aplikasi ini	
10	Saya merasa tertarik untuk menggunakan aplikasi ini	

5.3.2. Daftar Penguji Perangkat Lunak

Pada subbab ini ditunjukkan daftar pengguna yang bertindak sebagai penguji coba aplikasi yang dibangun. Dalam pengujian ini tidak terdapat kriteria atau keahlian khusus yang harus dimiliki pengguna karena aplikasi ini ditujukan kepada begbagai kalangan pengguna baik yang bisa bermain musik maupun tidak dapat bermain musik. Daftar nama penguji aplikasi ini dapat dilihat pada Tabel 5.9.

Tabel 5.9 Daftar Penguji Perangkat Lunak

No	Nama	Pekerjaan
1	Luthfi F. Soehadak	Mahasiswa Teknik Informatika ITS
2	Fajar Setiawan	Mahasiswa Teknik Informatika ITS
3	Dimas Widdy	Mahasiswa Teknik Informatika ITS
4	Oshi P. Gusman	Mahasiswa Teknik Informatika ITS
5	Ikrom Aulia Fahdi	Mahasiswa Teknik Informatika ITS
6	Fahmy Thoriqul Haq	Mahasiswa Teknik Informatika ITS
7	Selinda Anggia Devi	Mahasiswa Farmasi Universitas Airlangga
8	Bayu Mahardika	Mahasiswa Teknik Kelautan ITS

5.3.3. Hasil Pengujian Pengguna

Uji coba yang dilakukan terhadap beberapa pengguna memiliki beberapa aspek yang dipisahkan berdasarkan antarmukq, *immersivity*, dan tingkat kenyamanan. Sistem penilaian didasarkan pada skala penghitungan satu sampai enam dimana skala satu menunjukkan nilai terendah dan skala enam menunjukkan skala

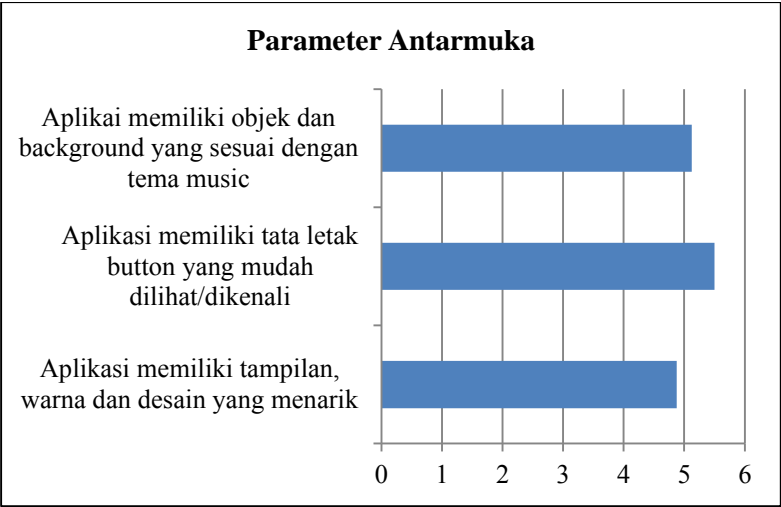
tertinggi. Penilaian akhir kemudian dilakukan dengan menghitung berapa banyak penguji yang memilih suatu skala tertentu dan kemudian dicari nilai rata-ratanya. Hasil uji coba dipaparkan secara lengkap dengan disertai tabel yang dapat dilihat pada Tabel 5.10 dan Tabel 5.11.

Tabel 5.10 Hasil Pengujian Pengguna

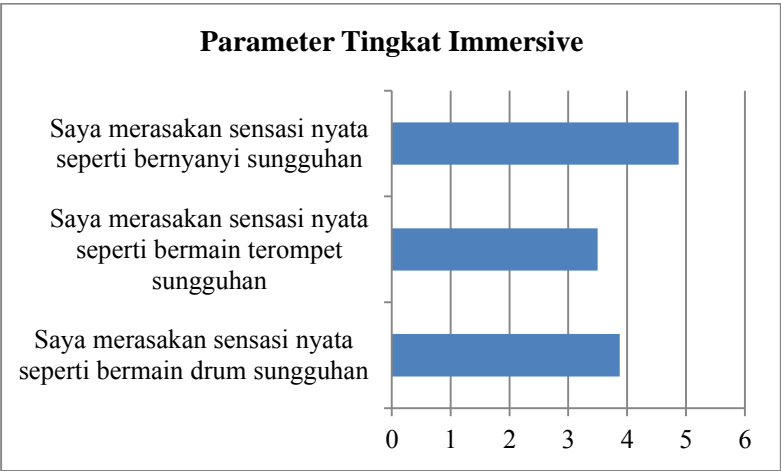
No	Pernyataan	Penilaian						Rata-Rata
		1	2	3	4	5	6	
Parameter Anatarmuka								
1	Aplikasi memiliki tampilan, warna dan desain yang menarik	0	0	1	1	4	2	4,87
2	Aplikasi memiliki tata letak <i>button</i> yang mudah dilihat/dikenali	0	0	0	0	4	4	5,50
3	Aplikai memiliki objek dan background yang sesuai dengan tema music	0	0	2	0	1	5	5,12
Parameter Immersivity								
4	Saya merasakan sensasi nyata seperti bermain drum sungguhan	0	1	1	5	0	1	3,87
5	Saya merasakan sensasi nyata seperti bermain terompet sungguhan	1	0	4	1	1	1	3,50
6	Saya merasakan sensasi nyata seperti bernyanyi sungguhan	0	0	0	2	5	1	4,87
Parameter Kenyamanan								
7	Aplikasi dapat berjalan lancar tanpa adanya <i>lag</i> dan <i>crash</i>	0	0	0	3	5	0	4,62
8	Saya merasa terbantu dengan adanya tutorial yang disediakan	0	0	1	1	5	1	4,75
9	Saya merasa nyaman selama menggunakan aplikasi ini	0	0	3	3	2	0	3,87
10	Saya merasa tertarik untuk menggunakan aplikasi ini	0	1	1	0	6	0	4,37

Tabel 5.11 Hasil Akhir Pengujian Pengguna

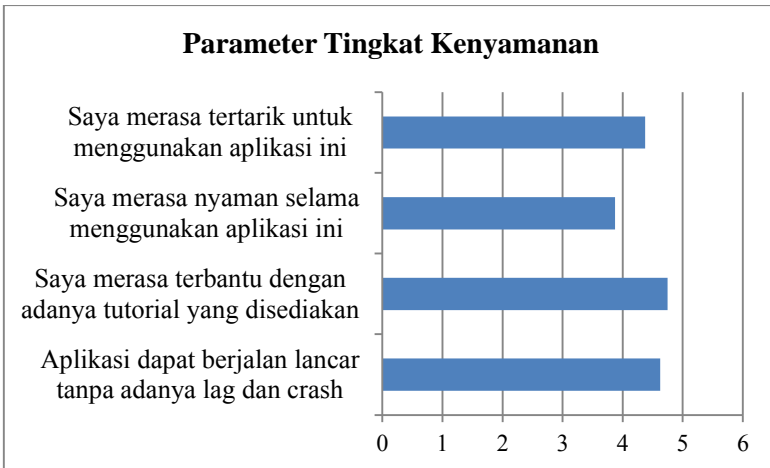
No	Pernyataan	Rata-Rata	Total	Total (%)
Parameter Antarmuka				
1	Aplikasi memiliki tampilan, warna dan desain yang menarik	4,87	5,16	86%
2	Aplikasi memiliki tata letak <i>button</i> yang mudah dilihat/dikenali	5,50		
3	Aplikai memiliki objek dan background yang sesuai dengan tema music	5,12		
Parameter Immersivity				
4	Saya merasakan sensasi nyata seperti bermain drum sungguhan	3,87	4,08	68%
5	Saya merasakan sensasi nyata seperti bermain terompet sungguhan	3,50		
6	Saya merasakan sensasi nyata seperti bernyanyi sungguhan	4,87		
Parameter Kenyamanan				
7	Aplikasi dapat berjalan lancar tanpa adanya <i>lag</i> dan <i>crash</i>	4,62	4,40	73,4%
8	Saya merasa terbantu dengan adanya tutorial yang disediakan	4,75		
9	Saya merasa nyaman selama menggunakan aplikasi ini	3,87		
10	Saya merasa tertarik untuk menggunakan aplikasi ini	4,37		



Gambar 5.11 Grafik Penilaian Antarmuka



Gambar 5.12 Grafik Penilaian Tingkat *Immersive*



Gambar 5.13 Grafik Penilaian Tingkat Kenyamanan

5.3.4. Kritik dan Saran Pengguna

Dalam memberikan penilaian dan tanggapan, pengguna diberikan kuesioner pengujian perangkat lunak. Kuesioner pengujian perangkat lunak ini terdapat bagian kritik dan saran untuk perbaikan fitur kedepannya. Kritik dan saran pengguna dapat dilihat pada Tabel 5.12.

Tabel 5.12 Kritik dan Saran Pengguna

No	Nama	Kritik dan Saran
1	Luthfi F. Soehadak	Berikan tutorial cara memainkan nada pada terompet, Agak sulit dalam memainkan drum karena terdapat perbedaan ketinggian pada tiap elemen drum
2	Fajar Setiawan	Aplikasi yang dikembangkan cukup menarik, hanya saja sensitifitas penggunaan leap motion kurang maksimal

3	Dimas Widdy	Terdapat beberapa delay namun permainan masih dapat diimplementasikan
4	Oshi P. Gusman	Dipermudah penggunaannya, dibuat lebih familiar
5	Ikrom Aulia Fahdi	Alat musik ditambahkan, UI diperbaiki
6	Fahmy Thoriqul Haq	Sangat susah memposisikan stik sesuai dengan posisi drum nya.
7	Selinda Anggia Devi	Alat musik yang dapat dimainkan diperbanyak
8	Bayu Mahardika	Posisi drumnya harus dikira-kira dahulu sehingga tidak dapat dimainkan secara cepat.

5.4. Evaluasi Pengujian

Berdasarkan hasil pengujian aplikasi, dan pengujian pengguna yang telah dijelaskan pada subbab sebelumnya, maka didapatkan evaluasi sebagai berikut:

1. Aplikasi berhasil mengeluarkan suara masing-masing alat drum. Hal ini dibuktikan pada pengujian UJ-P-01.
2. Aplikasi berhasil mengeluarkan suara terompet sesuai dengan *not* yang dimainkan. Hal ini dibuktikan pada pengujian UJ-P-02.
3. Aplikasi berhasil menangkap suara pengguna melalui mikrofon. Hal ini dibuktikan pada pengujian UJ-P-03.
4. Aplikasi berhasil merekam setiap alat yang dimainkan. Hal ini dibuktikan pada pengujian UJ-P-04.
5. Aplikasi berhasil menampilkan tutorial/bantuan untuk pengguna. Hal ini dibuktikan pada pengujian UJ-P-05.
6. Desain antarmuka aplikasi sudah sangat baik hal ini terbukti dari presentase hasil penilaian antarmuka yang mencapai 86%
7. Tingkat *immersive* pada aplikasi ini kurang baik hal ini terbukti dari presentase tingkat *immersive* yang hanya

mencapai 68%. Nilai rata-rata dalam memainkan drum adalah 3,87 dari nilai maksimal 6,0 dikarenakan pengguna harus menebak posisi drum terlebih dahulu sebelum memainkannya. Nilai rata-rata dalam memainkan terompet adalah 3,50 dari nilai maksimal 6,0 dikarenakan deteksi pergerakan jari kurang sensitive menurut pengguna. Hal ini dapat dibuktikan pada Tabel 5.12 Kritik dan Saran Pengguna. Kedua nilai rata-rata tersebut yang menjadikan nilai presentase *immersive* hanya mencapai 68%.

8. Tingkat kenyamanan pengguna selama menggunakan aplikasi ini memiliki presentase 73,4%. Beberapa pengguna merasa kurang nyaman karena dalam memainkan drum harus menebak terlebih dahulu. Hal ini dibuktikan pada Tabel 5.12 Kritik dan Saran Pengguna.

LAMPIRAN



KUESIONER TUGAS AKHIR – 5112100085 ANGGA SAPUTRA D. W.

RANCANG BANGUN APLIKASI *VIRTUAL MUSIC COMPOSER*, SEBAGAI MEDIA MENCIPTAKAN SEBUAH LAGU MENGGUNAKAN LEAP MOTION CONTROLLER

Identitas Responden

Nama Lengkap : Luthfi F Soehadlak Usia : 22 Tahun
 Pekerjaan : Managesswa Jenis Kelamin : P
 Keahlian Alat Musik : gitar

A. KARAKTERISTIK RESPONDEN

1. Pernahkah anda menggunakan leap motion controller?
 a. ☒ Tidak Pernah c. 2 kali e. 4 kali
 b. 1 kali d. 3 kali f. ≥ 5 kali
2. Seberapa sering anda memainkan drum dalam 1 bulan?
 a. ☒ Tidak Pernah c. 2 kali e. 4 kali
 b. 1 kali d. 3 kali f. ≥ 5 kali
3. Seberapa sering anda memainkan terompet dalam 1 bulan?
 a. ☒ Tidak Pernah c. 2 kali e. 4 kali
 b. 1 kali d. 3 kali f. ≥ 5 kali

B. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (✓)

SS = Sangat Setuju S = Setuju AS = Agak Setuju
 KS = Kurang Setuju TS = Tidak Setuju STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	KS	CS	S	SS
1	Aplikasi memiliki tampilan, warna dan desain yang menarik					✓	
2	Aplikasi memiliki tata letak <i>button</i> yang mudah dilihat/dikenali						✓
3	Aplikasi memiliki objek dan background sesuai dengan tema musik					✓	
Parameter Immersive							
4	Saya merasakan sensasi nyata seperti bermain drum sungguhan				✓		
5	Saya merasakan sensasi nyata seperti bermain terompet sungguhan			✓			
6	Saya merasakan sensasi nyata seperti bernyanyi sungguhan					✓	
Parameter Kenyamanan							
7	Aplikasi dapat berjalan lancar tanpa adanya <i>lag</i> dan <i>crash</i>				✓		
8	Saya merasa terbantu dengan adanya tutorial yang disediakan					✓	
9	Saya merasa nyaman selama menggunakan aplikasi ini			✓			
10	Saya merasa tertarik untuk menggunakan aplikasi ini		✓				

C. KRITIK DAN SARAN

1. Memberikan tutorial cara memainkan nada pada terompet
2. agak sulit dalam memainkan drum karena terdapat perbedaan ketinggian pada tiap elemen drum.

Surabaya, 7 Juni 2016

Gambar A.0.1 Kuesioner Responden Pertama



KUESIONER TUGAS AKHIR – 5112100085 ANGGA SAPUTRA D. W.

RANCANG BANGUN APLIKASI *VIRTUAL MUSIC COMPOSER*, SEBAGAI MEDIA MENCIPTAKAN SEBUAH LAGU MENGGUNAKAN LEAP MOTION CONTROLLER

Identitas Responden

Nama Lengkap : Fajar Setiawan Usia : 21 Tahun
 Pekerjaan : Mahasiswa Jenis Kelamin : L P
 Keahlian Alat Musik : -

A. KARAKTERISTIK RESPONDEN

- Pernahkah anda menggunakan leap motion controller?
 - Tidak Pernah
 - 1 kali
 - 2 kali
 - 3 kali
 - 4 kali
 - ≥ 5 kali
- Seberapa sering anda memainkan drum dalam 1 bulan?
 - Tidak Pernah
 - 1 kali
 - 2 kali
 - 3 kali
 - 4 kali
 - ≥ 5 kali
- Seberapa sering anda memainkan terompet dalam 1 bulan?
 - Tidak Pernah
 - 1 kali
 - 2 kali
 - 3 kali
 - 4 kali
 - ≥ 5 kali

B. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (✓)

SS = Sangat Setuju

S = Setuju

AS = Agak Setuju

KS = Kurang Setuju

TS = Tidak Setuju

STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	KS	CS	S	SS
1	Aplikasi memiliki tampilan, warna dan desain yang menarik					✓	
2	Aplikasi memiliki tata letak <i>button</i> yang mudah dilihat/dikenali						✓
3	Aplikasi memiliki objek dan background sesuai dengan tema musik						✓
Parameter Immersive							
4	Saya merasakan sensasi nyata seperti bermain drum sungguhan			✓			
5	Saya merasakan sensasi nyata seperti bermain terompet sungguhan			✓			
6	Saya merasakan sensasi nyata seperti bernyanyi sungguhan					✓	
Parameter Kenyamanan							
7	Aplikasi dapat berjalan lancar tanpa adanya <i>lag</i> dan <i>crash</i>						✓
8	Saya merasa terbantu dengan adanya tutorial yang disediakan						✓
9	Saya merasa nyaman selama menggunakan aplikasi ini				✓		
10	Saya merasa tertarik untuk menggunakan aplikasi ini						✓

C. KRITIK DAN SARAN

Aplikasi yang dikembangkan cukup menarik hanya saja sensasi nyata dengan menggunakan leap motion kurang maksimal. Pengguna seperti saya masih bingung dalam menggunakan cara mengoperasikan dengan sensor leap motion.

Surabaya, 07 Juni 2016

Gambar A.0.2 Kuesioner Responden Kedua



KUESIONER TUGAS AKHIR – 5112100085 ANGGA SAPUTRA D. W.

RANCANG BANGUN APLIKASI *VIRTUAL MUSIC COMPOSER*, SEBAGAI MEDIA MENCIPTAKAN
SEBUAH LAGU MENGGUNAKAN LEAP MOTION CONTROLLER

Identitas Responden

Nama Lengkap : Dimas Waddy Usia : 22 Tahun
Pekerjaan : Mahasiswa Jenis Kelamin : (L) P
Keahlian Alat Musik : Gitar

A. KARAKTERISTIK RESPONDEN

1. Pernahkah anda menggunakan leap motion controller?
 - a. Tidak Pernah
 - b. 1 kali
 - c. 2 kali
 - d. 3 kali
 - e. 4 kali
 - f. >= 5 kali
2. Seberapa sering anda memainkan drum dalam 1 bulan?
 - a. Tidak Pernah
 - b. 1 kali
 - c. 2 kali
 - d. 3 kali
 - e. 4 kali
 - f. >= 5 kali
3. Seberapa sering anda memainkan terompet dalam 1 bulan?
 - a. Tidak Pernah
 - b. 1 kali
 - c. 2 kali
 - d. 3 kali
 - e. 4 kali
 - f. >= 5 kali

B. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (v)

SS = Sangat Setuju

S = Setuju

AS = Agak Setuju

KS = Kurang Setuju

TS = Tidak Setuju


STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	KS	CS	S	SS
1	Aplikasi memiliki tampilan, warna dan desain yang menarik						✓
2	Aplikasi memiliki tata letak <i>button</i> yang mudah dilihat/dikenali						✓
3	Aplikasi memiliki objek dan background sesuai dengan tema musik						✓
Parameter Immersive							
4	Saya merasakan sensasi nyata seperti bermain drum sungguhan						✓
5	Saya merasakan sensasi nyata seperti bermain terompet sungguhan					✓	
6	Saya merasakan sensasi nyata seperti bernyanyi sungguhan					✓	
Parameter Kenyamanan							
7	Aplikasi dapat berjalan lancar tanpa adanya <i>lag</i> dan <i>crash</i>				✓		
8	Saya merasa terbantu dengan adanya tutorial yang disediakan						✓
9	Saya merasa nyaman selama menggunakan aplikasi ini				✓		
10	Saya merasa tertarik untuk menggunakan aplikasi ini					✓	

C. KRITIK DAN SARAN

terdapat beberapa bug delay, namun permainan
masih bagus di implementasikan

Surabaya, 7 Juli 2016


Dimas Waddy

Gambar A.03 Kuesioner Responden Ketiga


KUESIONER TUGAS AKHIR – 5112100085 ANGGA SAPUTRA D. W.
RANCANG BANGUN APLIKASI VIRTUAL MUSIC COMPOSER, SEBAGAI MEDIA MENCIPTAKAN SEBUAH LAGU MENGGUNAKAN LEAP MOTION CONTROLLER
Identitas Responden

Nama Lengkap : OSU R. GUSMAN Usia : 22.. Tahun
 Pekerjaan : MAHASISWA Jenis Kelamin : L / ♂
 Keahlian Alat Musik :

A. KARAKTERISTIK RESPONDEN

1. Pernahkah anda menggunakan leap motion controller?
 - a. ☒ Tidak Pernah
 - b. ☐ 1 kali
 - c. ☐ 2 kali
 - d. ☐ 3 kali
 - e. ☐ 4 kali
 - f. ☐ >= 5 kali
2. Seberapa sering anda memainkan drum dalam 1 bulan?
 - a. ☒ Tidak Pernah
 - b. ☐ 1 kali
 - c. ☐ 2 kali
 - d. ☐ 3 kali
 - e. ☐ 4 kali
 - f. ☐ >= 5 kali
3. Seberapa sering anda memainkan terompet dalam 1 bulan?
 - a. ☒ Tidak Pernah
 - b. ☐ 1 kali
 - c. ☐ 2 kali
 - d. ☐ 3 kali
 - e. ☐ 4 kali
 - f. ☐ >= 5 kali

B. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (✓)

SS = Sangat Setuju

S = Setuju

AS = Agak Setuju

KS = Kurang Setuju

TS = Tidak Setuju

STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	KS	CS	S	SS
1	Aplikasi memiliki tampilan, warna dan desain yang menarik					<input checked="" type="checkbox"/>	
2	Aplikasi memiliki tata letak <i>button</i> yang mudah dilihat/dikenali					<input checked="" type="checkbox"/>	
3	Aplikasi memiliki objek dan background sesuai dengan tema musik						<input checked="" type="checkbox"/>
Parameter Immersive							
4	Saya merasakan sensasi nyata seperti bermain drum sungguhan				<input checked="" type="checkbox"/>		
5	Saya merasakan sensasi nyata seperti bermain terompet sungguhan						<input checked="" type="checkbox"/>
6	Saya merasakan sensasi nyata seperti bernyanyi sungguhan				<input checked="" type="checkbox"/>		
Parameter Kenyamanan							
7	Aplikasi dapat berjalan lancar tanpa adanya <i>lag</i> dan <i>crash</i>				<input checked="" type="checkbox"/>		
8	Saya merasa terbantu dengan adanya tutorial yang disediakan				<input checked="" type="checkbox"/>		
9	Saya merasa nyaman selama menggunakan aplikasi ini				<input checked="" type="checkbox"/>		
10	Saya merasa tertarik untuk menggunakan aplikasi ini					<input checked="" type="checkbox"/>	

C. KRITIK DAN SARAN

dipermudah pengoperasiannya, dibuat lebih familiar.

Surabaya, 7 Juni..... 2016

Agus

Gambar A.0.4 Kuesioner Responden Keempat



KUESIONER TUGAS AKHIR – 5112100085 ANGGA SAPUTRA D. W.

RANCANG BANGUN APLIKASI *VIRTUAL MUSIC COMPOSER*, SEBAGAI MEDIA MENCIPTAKAN SEBUAH LAGU MENGGUNAKAN LEAP MOTION CONTROLLER

Identitas Responden

Nama Lengkap : Heron Ajia Fahri Usia : 22 Tahun
 Pekerjaan : Mahasiswa Jenis Kelamin : L P
 Keahlian Alat Musik : Gitar, Bass

A. KARAKTERISTIK RESPONDEN

- Pernahkah anda menggunakan leap motion controller?
☒ a. Tidak Pernah c. 2 kali e. 4 kali
 b. 1 kali d. 3 kali f. >= 5 kali
- Seberapa sering anda memainkan drum dalam 1 bulan?
 a. Tidak Pernah c. 2 kali e. 4 kali
☒ b. 1 kali d. 3 kali f. >= 5 kali
- Seberapa sering anda memainkan terompet dalam 1 bulan?
 a. Tidak Pernah c. 2 kali e. 4 kali
☒ b. 1 kali d. 3 kali f. >= 5 kali

B. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (v)

SS = Sangat Setuju

S = Setuju

AS = Agak Setuju

KS = Kurang Setuju

TS = Tidak Setuju

STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	KS	CS	S	SS
1	Aplikasi memiliki tampilan, warna dan desain yang menarik				✓		
2	Aplikasi memiliki tata letak <i>button</i> yang mudah dilihat/dikenali					✓	
3	Aplikasi memiliki objek dan background sesuai dengan tema musik			✓			
Parameter Immersive							
4	Saya merasakan sensasi nyata seperti bermain drum sungguhan				✓		
5	Saya merasakan sensasi nyata seperti bermain terompet sungguhan			✓			
6	Saya merasakan sensasi nyata seperti bernyanyi sungguhan				✓		
Parameter Kenyamanan							
7	Aplikasi dapat berjalan lancar tanpa adanya <i>lag</i> dan <i>crash</i>						✓
8	Saya merasa terbantu dengan adanya tutorial yang disediakan						✓
9	Saya merasa nyaman selama menggunakan aplikasi ini						✓
10	Saya merasa tertarik untuk menggunakan aplikasi ini						✓

C. KRITIK DAN SARAN

Alat Musiknya kalau bisa ditambahkan ul interface di perangkat

Surabaya, 7-06-2016

Heron AP

Gambar A.0.5 Kuesioner Responden Kelima



KUESIONER TUGAS AKHIR – 5112100085 ANGGA SAPUTRA D. W.

RANCANG BANGUN APLIKASI *VIRTUAL MUSIC COMPOSER*, SEBAGAI MEDIA MENCIPTAKAN SEBUAH LAGU MENGGUNAKAN LEAP MOTION CONTROLLER

Identitas Responden

Nama Lengkap : Fahmy Thorau Hay
 Pekerjaan : Mahasiswa
 Keahlian Alat Musik : Gitar - bassener
 Usia : 19 Tahun
 Jenis Kelamin : P

A. KARAKTERISTIK RESPONDEN

- Pernahkah anda menggunakan leap motion controller?
 - Tidak Pernah
 - 1 kali
 - 2 kali
 - 3 kali
 - 4 kali
 - ≥ 5 kali
- Seberapa sering anda memainkan drum dalam 1 bulan?
 - Tidak Pernah
 - 1 kali
 - 2 kali
 - 3 kali
 - 4 kali
 - ≥ 5 kali
- Seberapa sering anda memainkan terompet dalam 1 bulan?
 - Tidak Pernah
 - 1 kali
 - 2 kali
 - 3 kali
 - 4 kali
 - ≥ 5 kali

B. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (v)

SS = Sangat Setuju

S = Setuju

AS = Agak Setuju

KS = Kurang Setuju

TS = Tidak Setuju

STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	KS	CS	S	SS
1	Aplikasi memiliki tampilan, warna dan desain yang menarik			✓			
2	Aplikasi memiliki tata letak <i>button</i> yang mudah dilihat/dikenali					✓	
3	Aplikasi memiliki objek dan background sesuai dengan tema musik						✓
Parameter Immersive							
4	Saya merasakan sensasi nyata seperti bermain drum sungguhan		✓				
5	Saya merasakan sensasi nyata seperti bermain terompet sungguhan	✓					
6	Saya merasakan sensasi nyata seperti bernyanyi sungguhan				✓		
Parameter Kenyamanan							
7	Aplikasi dapat berjalan lancar tanpa adanya <i>lag</i> dan <i>crash</i>					✓	
8	Saya merasa terbantu dengan adanya tutorial yang disediakan					✓	
9	Saya merasa nyaman selama menggunakan aplikasi ini			✓			
10	Saya merasa tertarik untuk menggunakan aplikasi ini			✓			

C. KRITIK DAN SARAN

- Sangat dan jangan menyerah
 ➤ Sangat perlu mengetahui cara bermain drum dan tata letaknya
 karena tidak berpengalaman

Surabaya, 1 Juni 2016

Gambar A.0.6 Kuesioner Responden Keenam


KUESIONER TUGAS AKHIR – 5112100085 ANGGA SAPUTRA D. W.
RANCANG BANGUN APLIKASI VIRTUAL MUSIC COMPOSER, SEBAGAI MEDIA MENCIPTAKAN SEBUAH LAGU MENGGUNAKAN LEAP MOTION CONTROLLER
Identitas Responden

Nama Lengkap : Selinda Anggia Dewi Usia : 21 Tahun
 Pekerjaan : Mahasiswa Jenis Kelamin : L (P)
 Keahlian Alat Musik : -

A. KARAKTERISTIK RESPONDEN

- Pernahkah anda menggunakan leap motion controller?
☒ a. Tidak Pernah c. 2 kali e. 4 kali
 b. 1 kali d. 3 kali f. ≥ 5 kali
- Seberapa sering anda memainkan drum dalam 1 bulan?
☒ a. Tidak Pernah c. 2 kali e. 4 kali
 b. 1 kali d. 3 kali f. ≥ 5 kali
- Seberapa sering anda memainkan terompet dalam 1 bulan?
☒ a. Tidak Pernah c. 2 kali e. 4 kali
 b. 1 kali d. 3 kali f. ≥ 5 kali

B. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (✓)

SS = Sangat Setuju

S = Setuju

AS = Agak Setuju

KS = Kurang Setuju

TS = Tidak Setuju

STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	KS	CS	S	SS
1	Aplikasi memiliki tampilan, warna dan desain yang menarik					✓	
2	Aplikasi memiliki tata letak <i>button</i> yang mudah dilihat/dikenali					✓	
3	Aplikasi memiliki objek dan background sesuai dengan tema musik			✓			
Parameter Immersive							
4	Saya merasakan sensasi nyata seperti bermain drum sungguhan				✓		
5	Saya merasakan sensasi nyata seperti bermain terompet sungguhan			✓			
6	Saya merasakan sensasi nyata seperti bernyanyi sungguhan					✓	
Parameter Kenyamanan							
7	Aplikasi dapat berjalan lancar tanpa adanya <i>lag</i> dan <i>crash</i>				✓	✓	
8	Saya merasa terbantu dengan adanya tutorial yang disediakan					✓	
9	Saya merasa nyaman selama menggunakan aplikasi ini					✓	
10	Saya merasa tertarik untuk menggunakan aplikasi ini					✓	

C. KRITIK DAN SARAN

Alat musik yang ~~dimainkan~~ dapat dimainkan diperbanyak

Surabaya, 7 Juni 2016

Selinda Anggia Dewi

Gambar A.0.7 Kuesioner Responden Ketujuh



KUESIONER TUGAS AKHIR – 5112100085 ANGGA SAPUTRA D. W.

RANCANG BANGUN APLIKASI VIRTUAL MUSIC COMPOSER, SEBAGAI MEDIA MENCIPTAKAN SEBUAH LAGU MENGGUNAKAN LEAP MOTION CONTROLLER

Identitas Responden

Nama Lengkap : Bayu Mahardika Usia : 22 Tahun
 Pekerjaan : Mahasiswa Jenis Kelamin : P
 Keahlian Alat Musik : Drum

A. KARAKTERISTIK RESPONDEN

- Pernahkah anda menggunakan leap motion controller?
 - Tidak Pernah
 - 1 kali
 - 2 kali
 - 3 kali
 - 4 kali
 - ≥ 5 kali
- Seberapa sering anda memainkan drum dalam 1 bulan?
 - Tidak Pernah
 - 1 kali
 - 2 kali
 - 3 kali
 - 4 kali
 - ≥ 5 kali
- Seberapa sering anda memainkan terompet dalam 1 bulan?
 - Tidak Pernah
 - 1 kali
 - 2 kali
 - 3 kali
 - 4 kali
 - ≥ 5 kali

B. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah ini dengan menggunakan tanda (v)

SS = Sangat Setuju

S = Setuju

AS = Agak Setuju

KS = Kurang Setuju

TS = Tidak Setuju

STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	KS	CS	S	SS
1	Aplikasi memiliki tampilan, warna dan desain yang menarik						✓
2	Aplikasi memiliki tata letak <i>button</i> yang mudah dilihat/dikenali						✓
3	Aplikasi memiliki objek dan background sesuai dengan tema musik						✓
Parameter Immersive							
4	Saya merasakan sensasi nyata seperti bermain drum sungguhan				✓		
5	Saya merasakan sensasi nyata seperti bermain terompet sungguhan					✓	
6	Saya merasakan sensasi nyata seperti bernyanyi sungguhan						
Parameter Kenyamanan							
7	Aplikasi dapat berjalan lancar tanpa adanya <i>lag</i> dan <i>crash</i>					✓	
8	Saya merasa terbantu dengan adanya tutorial yang disediakan			✓			
9	Saya merasa nyaman selama menggunakan aplikasi ini			✓			
10	Saya merasa tertarik untuk menggunakan aplikasi ini					✓	

C. KRITIK DAN SARAN

posisi drum harus dikera - kera dahulu, sehingga tidak dapat
 dimainkan secara cepat

Surabaya, 7 Juni 2016

[Signature]

Gambar A.08 Kuesioner Responden Kedelapan

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan Tugas Akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1. Kesimpulan

Dari hasil selama proses perancangan, implementasi, serta pengujian dapat diambil kesimpulan sebagai berikut:

1. Aplikasi dapat mendeteksi gerakan pukulan stik drum untuk memainkan drum yang ditangkap oleh leap motion controller.
2. Aplikasi dapat mendeteksi gerakan jari tangan untuk memainkan *not* terompet yang ditangkap oleh leap motion controller.
3. Aplikasi dapat mendeteksi tiupan untuk membunyikan terompet melalui *pitch value* yang didapatkan dari mikrofon.
4. Aplikasi dapat merekam suara drum, terompet dan mikrofon dengan menggunakan *list*.
5. Aplikasi telah berhasil menciptakan sebuah lagu melalui simulasi memainkan drum, terompet dan bernyanyi.
6. Leap Motion tidak dapat mendeteksi stik drum dalam keadaan menyilang.
7. Simulasi memainkan drum dapat direalisasikan namun pemanfaatan leap motion controller dalam simulasi bermain drum kurang tepat dikarenakan leap motion memiliki keterbatasan jangkauan yang hanya mencapai 30cm kearah sumbu *y*. Drum juga memiliki ketinggian pada masing-masing bagian yang berbeda sehingga pengguna juga harus menebak ketinggian dari bagian drum sebelum memukul.

6.2. Saran

Berikut saran-saran untuk pengembangan dan perbaikan sistem di masa yang akan datang. Diantaranya adalah sebagai berikut:

1. Sebagai alternatif lain untuk pendeteksi tiupan dapat dilakukan dengan metode *noise detection* namun harus menggunakan mikrofon yang terpasang pada *smartphone*.
2. Untuk mendapatkan jangkauan leap motion controller yang lebih lebar dapat menggabungkan dua leap motion sehingga jangkauan memainkan drum lebih lebar.

DAFTAR PUSTAKA

- [1] E. S. Silva, J. A. O. d. Abreu, J. H. P. d. Almeida, V. Teichrieb and G. L. Ramalho, "A Preliminary Evaluation of the Leap Motion Sensor as Controller of New Digital Musical Instruments," in *14^o Simpósio Brasileiro de Computação*, Brasília, Sociedade Brasileira de Computação – SBC, 2013, pp. 59-70.
- [2] I. Sommerville, *Software Engineering* (9th ed.), Harlow, England: Pearson Education, 2010.
- [3] Sokanu, "What does a Composer do?," [Online]. Available: <https://www.sokanu.com/careers/composer/>. [Accessed 15 July 2016].
- [4] A. Nalwan, *Pemrograman Animasi dan Game Profesional*, Jakarta: Elex Media Komputindo, 1998.
- [5] V. Llopis, "Introducing the Skeletal Tracking Model," *The Motion Report*, 6 May 2013. [Online]. Available: <http://themotionreport.com/evolution-of-leap-motion-controller/>. [Accessed 25 May 2016].
- [6] A. Colgan, "Introducing the Skeletal Tracking Model," *Leap Motion, Inc*, 9 August 2014. [Online]. Available: <http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>. [Accessed 25 May 2016].
- [7] L. Motion, "Leap Motion API Classes," *Leap Motion, Inc*, 21 January 2012. [Online]. Available: https://developer.leapmotion.com/documentation/v2/unity/api/Leap_Classes.html#leap-motion-api-classes. [Accessed 25 May 2016].
- [8] Unity, "Game engine, tools and multi platform," *Unity*, [Online]. Available: <http://unity3d.com/unity>. [Accessed 9 april 2014].

- [9] Microsoft, "Microsoft studio- development tools," Microsoft, [Online]. Available: <http://www.visualstudio.com/>. [Accessed 9 april 2014].
- [10] Blender, "about-blender," Blender, [Online]. Available: <http://www.blender.org/about/>. [Accessed 10 april 2014].
- [11] N. Software, "NCH Software," NCH Software, Inc., 02 September 2015. [Online]. Available: <http://www.nch.com.au/wavepad/>. [Accessed 03 May 2016].
- [12] I. Ilascu, "Softpedia," Bandisoft, 9 April 2011. [Online]. Available: <http://www.softpedia.com/reviews/windows/Bandicam-Review-194089.shtml>. [Accessed 16 May 2016].
- [13] E. Alwando, Rancang Bangun Aplikasi Virtual Indonesian Musical Instrument (VIMi.) Sebagai Media Interaktif dalam Penggunaan Alat Musik Tradisional Menggunakan Leap Motion Controller, Surabaya: Ruang Baca Teknik Computer ITS, 2015.
- [14] P. E. Bawono, Realitas Virtual Pertunjukan Wayang Golek Dengan Menggunakan Leap Motion Sebagai Controller Model Wayang 3D, Surabaya: Ruang Baca Teknik Computer ITS, 2015.
- [15] Rodrigo, "Real Drum," Rodrigo Kolb Apps, 27 February 2016. [Online]. Available: <https://play.google.com/store/apps/details?id=br.com.rodrigokolb.realdrum&hl=in>. [Accessed 15 July 2016].
- [16] B. Martinson, "Baritone/Euphonium," Beginband, 2001. [Online]. Available: <http://www.beginband.com/baritone.shtml>. [Accessed 15 July 2016].

BIODATA PENULIS



Penulis, Angga Saputra Dwi Wardana, lahir di kota Sidoarjo pada tanggal 31 Maret 1994. Penulis dibesarkan di kota Surabaya, Jawa Timur.

Penulis menempuh pendidikan formal di TK Hasanuddin Surabaya (1999-2000), SDN Kebraon II/437 Surabaya (2000-2006), SMPN 1 Surabaya (2006-2009), SMAN 5 Surabaya (2009-2012). Pada tahun 2012, penulis melanjutkan pendidikan S1 jurusan Teknik Informatika Fakultas Teknologi Informasi di Institut Teknologi Sepuluh Nopember Surabaya, Jawa Timur.

Di jurusan Teknik Informatika, penulis mengambil bidang minat Interaksi Grafika dan Seni atau biasa disingkat menjadi IGS dan memiliki ketertarikan di bidang *back end Web*, Augmented Reality, Realitas Virtual, dan pembuatan Game. Penulis aktif sebagai staf Departemen Kewirausahaan Minat dan Bakat Himpunan Mahasiswa Teknik Komputer periode kepengurusan 2013–2014 dan sebagai staf ahli Departemen Kewirausahaan Minat dan Bakat Himpunan Mahasiswa Teknik Komputer periode kepengurusan 2014–2015 serta kegiatan organisasi dan event mahasiswa lainnya. Penulis dapat dihubungi melalui alamat email angga.saputradwi@gmail.com